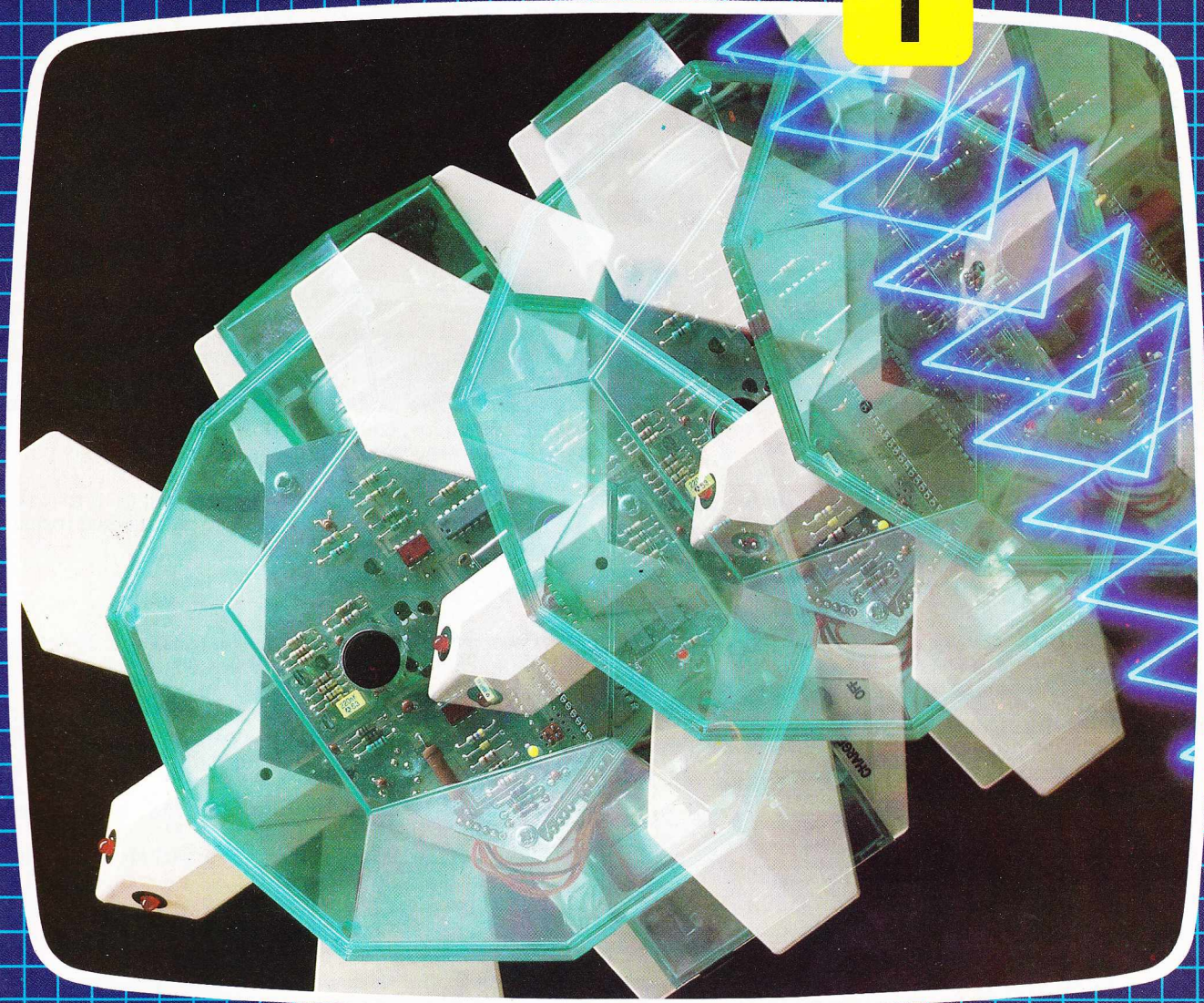
MICROCOMPUTADOR CURSO PRÁTICO

TIRE O MÁXIMO DO MICRO EM 30 SEMANAS

1



Linguagem LOGO • Lógica computacional
Processadores de texto

rioGráfica

MICROCOMPUTADOR CURSO PRÁTICO

APLICAÇÕES

A escrita na tela 17

PERIFÉRICOS

Escolha de arquivos 20

SOFTWARE

Lições e desafios 22

PROJETOS DE PROGRAMAS

A próxima letra 24

RAIO X

IBM PC 26

CIÊNCIA DA COMPUTAÇÃO

A álgebra das decisões 28

LINGUAGENS

Os passos da tartaruga 30

TÉCNICAS DE PROGRAMAÇÃO

Plano de ação 34

PERFIL

Revolução nos circuitos 36

TESTE

Depois de ler com atenção todo o fascículo, responda a estas perguntas. Veja a solução no próximo número, junto com um novo teste.

1) Como são gravados os dados dentro de um cartucho?
2) O que caracteriza o sistema de endereçamento do IBM PC?
3) O que são softwares integrados?
4) Quais as três operações básicas da lógica booleana?
5) Que linguagem deu origem ao LOGO?
6) Qual comando LOGO faz a tartaruga mover-se sem traçar linhas?

PLANO DA OBRA

MICROCOMPUTADOR - CURSO PRÁTICO é uma coleção de 31 fascículos de periodicidade semanal. Os trinta primeiros terão, cada um, vinte páginas internas (miolo) e quatro capas, além de uma introdução de dezesseis páginas acompanhando o fascículo 1. O fascículo 31 terá doze páginas e quatro capas, e incluirá o índice geral da obra. Os miolos, encadernados, constituem dois volumes de um curso prático de microcomputação, com seções indicadas por tarjas de diferentes cores.
As duas primeiras capas são descartáveis; as terceiras e quartas trazem programas de jogos. Encadernadas, formam um volume de cem páginas.

COMO ENCADERNAR

As capas duras, incluindo as guardas, estarão à venda simultaneamente com os fascículos 10 (**Volume 1**) e 25 (**Volume 2** e **Jogos**).
Volume 1 - Deve ser encadernado com os elementos dispostos nesta ordem: guardas; frontispício e introdução (publicados com o fascículo 1); miolos dos fascículos de 1 a 15; e guardas.
Volume 2 - Guardas; frontispício (que virá com o fascículo 31); miolos dos fascículos de 16 a 31; e guardas.
Jogos - Guardas; frontispício e sumário (que constituem a 3.ª e 4.ª capas do fascículo 31); 3.ª e 4.ª capas dos fascículos de 1 a 30; e guardas.

rioGráfica **Diretoria Executiva:** Oscar Neves; Filipe Zander; Nilo Sérgio de Almeida; Danilo Esteves Costa; **GRUPO EDUCAÇÃO E CULTURA - Planejamento e Comercialização - Diretor:** Roberto Combochi; **Gerente de Marketing:** Jaime Rodrigues; **Gerente de Produto:** José Mauro Porto; **Gerente do P.C.P.:** Aylton Menezes; **Gerente de Vendas:** Valter Aparecido Lopes; **Gerente de Promoção:** Rubens Barbosa; **Coordenadores de Promoção:** Paulo Cesar M. Seixas; Edgar Mello M. Neto; **Supervisor de Circulação:** Norberto Martin; **Supervisor de Operações:** Abel Pereira Pinto - **Editorial - Diretor:** João J. Noro; **Editor-chefe:** Mauricio Rittner; **Administração Editorial:** Heitor de Souza Paixão; **Editor:** Jesse Navarro; **Revisores:** Isis Augusta Loyolla; Cacilda Guerra; Luiz Vicente Vieira Filho; Marcos Domingos Agathão; Maria da Graça Mendonça Couto; **Diretor de Arte:** Anibal Monteiro; **Chefe de Arte:** Bonifácio Duardes Miranda; **Chefe de Estúdio:** José Yuji Kuribayashi; **Assistente de Arte:** Miguel Luis Escámez Simón - **Colaboradores - Editores-assistentes:** Isa Mara Lando; Luiz Carlos Pizarro Marin; **Texto:** Geraldo Carlos Nascimento; **Preparação de Texto:** Luiz Carlos Cardoso; **Tradução:** Cláudio Marcondes; Maria Clara Cescato; Patrícia de Paiva e Castro; Regina Amarante; Regina Carmo R. Silva; **Pesquisa Iconográfica:** Stella Maria Quentel; **Arte:** Toshio Morobosi; **Assessoria Técnica:** Derval Junqueira de Aquino; Lauro de Lauro; Luiz Henrique Alayon. As fotos não creditadas pertencem à obra original. Seguem-se os créditos das demais: Brasoft, Salvador de Rosa, Fernando Scavone.

© APSIF Copenhagen, 1984; © Orbis Publishing Ltd., 1984; © Editora Rio Gráfica Ltda., 1985, para a língua portuguesa.

Composição: ERG Ltda. e AM Produções Gráficas Ltda. **Impressão:**

NÚMEROS ATRASADOS - Você poderá comprar os exemplares que faltam em sua coleção, pelo preço do último fascículo posto à venda, em sua banca de jornal preferida. Caso não consiga obtê-los, faça sua solicitação por carta endereçada diretamente à Editora Rio Gráfica Ltda., rua Itapiru, 1209, Rio Comprido, Rio de Janeiro, CEP 20251. O atendimento será feito por via postal. Nas cidades do Rio de Janeiro e de São Paulo, as compras poderão ser efetuadas pessoalmente nos seguintes endereços: Rio de Janeiro - Rua Itapiru, 1209, Rio Comprido; São Paulo - Rua Frei Caneca, 1152, Consolação.

Distribuidor exclusivo para todo o Brasil: Fernando Chinaglia - Rua Teodoro da Silva, 907, Rio de Janeiro. Distribuidor para Portugal: Electroliber Lda. - Rua Prof. Reinaldo dos Santos, 1488, Lisboa.

A ESCRITA NA TELA

Texto impecável
Depois de lidos e corrigidos na tela, os textos são transcritos em papel por uma impressora acoplada ao micro. As modificações introduzidas na cópia impressa são executadas na tela, redigitando-se apenas as correções.

Por meio de programas que manipulam textos, muitas pessoas têm seu primeiro contato com os micros. Conheça aqui as mais importantes aplicações e os recursos oferecidos por esse tipo de software.

Imagine o que significa redigir um manual técnico, uma apostila, uma tese ou mesmo um artigo de jornal. É necessário fazer muitas correções para que o texto fique claro, correto, bem organizado e adequado ao espaço que deve ocupar. Palavras e frases inteiras são modificadas ou eliminadas e novos conceitos, inseridos. Talvez seja preciso trocar a ordem de sentenças, parágrafos e até páginas inteiras.

Como fazer tudo isso dispondo apenas de papel e lápis ou de uma simples máquina de escrever? Os roteiristas de cinema, que com freqüência alteram a ordem das tomadas, costumam lançar mão de tesoura e cola. Outros profissionais recorrem a datilógrafas para refazer os textos, num trabalho repetitivo, que conta com poucos recursos para ficar correto e com boa estética.

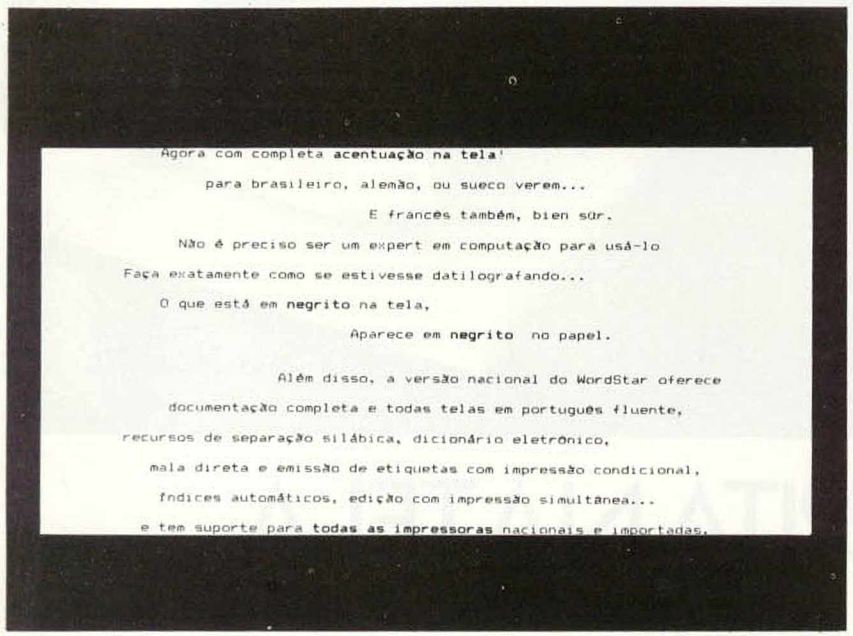
A difusão dos microcomputadores trouxe o auxílio dos processadores (ou editores) de texto na resolução desses problemas. São programas que manipulam desde uma única letra até muitas páginas escritas, facilitando o trabalho de redação e correção de textos comerciais, técnicos, jornalísticos, didáticos e literários, além de correspondência e anotações pessoais.

Na utilização do microcomputador, os programas combinam os recursos de máquina de escrever, tela de vídeo e meios magnéticos de armazenamento de dados. Os textos são redigidos e alterados com facilidade na tela. Mediante combinações de teclas, que variam conforme o software, emitem-se comandos para inserir, eliminar, alterar, mover ou copiar letras, palavras, sentenças, parágrafos ou páginas inteiras.

Pronto, o texto e armazenado. Em geral o usuário grava-o em disquetes, que ocupam mui-

to menos volume que um arquivo de textos impressos. Os textos armazenados são lidos e corrigidos em tela sempre que necessário e, após a correção, transcritos em papel por uma impressora acoplada ao micro. Novas modificações, anotadas na cópia impressa, são em seguida executadas na tela. Altera-se apenas o que está impróprio; isso praticamente dispensa a redigitação. A correção fica facilitada por não se trabalhar com o texto no papel, mas em meios magnéticos. Obtém-se, assim, uma cópia final correta e com aparência impecável.

Entre os recursos mais comuns dos softwares atuais contam-se: alinhamento automático do texto pela direita; margens e tabuladores ajustáveis; mostrador do número de página, de linha e de coluna; marca automática de fim de página; centralização de títulos; negrito, sublinhado, subscrito (como no 2 de "H_2O") e superescrito (como em "1.º"), indicados por símbolos especiais.



Recursos amplos
O WordStar, campeão de vendas americano, foi adaptado para o Brasil com a introdução de caracteres da grafia portuguesa. Esse programa mostra o texto na tela como será impresso no papel. Os recursos dos processadores de texto são amplos: alguns programas dispõem até de buffer para armazenamento: enquanto a impressora produz um texto, o micro está liberado para processar outro (abaixo).

Nos processadores mais modernos, o texto aparece na tela exatamente como ficará na página impressa; o cabeçalho e o rodapé são automáticos (uma vez definidos, imprimem-se em todas as páginas). Recursos para encontrar e substituir determinadas palavras ou seqüências de caracteres permitem trocar uma palavra por outra no texto inteiro, com um único comando.

Por serem bastante elaborados, os softwares processadores de texto ocupam muito lugar na memória. Além disso, para obter resultados significativos, é necessário que haja acesso direto (aleatório) aos textos —, ou seja, a fim de visualizar certa informação na tela, basta indicá-la no diretório ou índice do disquete; o sistema operacional se encarrega de localizá-la.

Por isso, os microcomputadores que têm pouca capacidade de memória e utilizam fitas cassete dispõem de processadores de texto com recursos limitados, pois a fita só permite o acesso seqüencial: para localizar uma informação é necessário passar por todas as outras que foram gravadas antes dela, orientando-se pelo contador numérico do gravador.

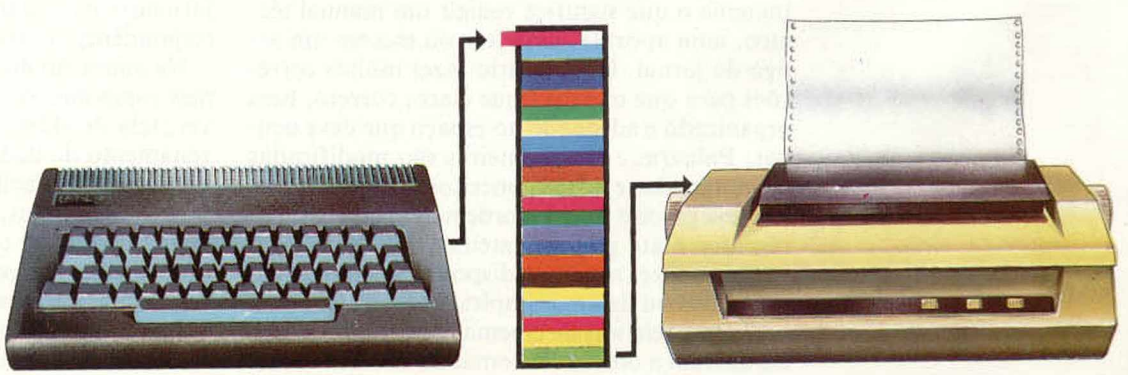
No entanto, há progressos na área dos portáteis. Dotados de visor de cristal líquido, que mostra algumas linhas de cada vez, tais equipamentos podem armazenar os textos em fita, cartucho ou microfita, sendo ideais para jornalistas e vendedores. Esses profissionais utilizam o micro como um bloco de anotações e transmitem o texto via modem e linhas telefônicas para sua sede, distante até centenas de quilômetros.

Outro passo importante na simplificação do processo é o uso de um editor de texto previamente gravado num circuito integrado, que fica, portanto, residente no micro.

Os principais

Ao comparar diversos softwares para processamento de texto, devem-se verificar suas características quanto a facilidade de uso, formatação do texto na tela, impressão e manipulação de arquivos. Alguns programas têm mais recursos mas também ocupam mais memória, custam mais caro e apresentam muitos comandos.

Entre os processadores de texto de maior uso no Brasil está, para a linha TRS-80, o Electric Pencil, um dos primeiros no gênero. A linha Apple oferece, entre outros, os modelos Apple Writer, Magic Window e WordStar, este um campeão de vendas americano.



Há ainda o Editex, desenvolvido no Brasil especialmente para o MicroEngenho, da Spectrum. Esse programa permite verificar a ortografia e a acentuação. Enquanto o usuário escreve, o programa aplica as 132 regras que tem armazenadas. Se detectar uma infração a essas regras, fará soar um alarma e mostrará a forma correta. Também desenvolvido no Brasil é o Redator, destinado aos equipamentos Itautec.

Os programas nacionais têm recursos para produzir as letras e os sinais usados em português, como a cedilha, o til e os acentos; contudo, o hardware também deve estar capacitado para mostrar esses sinais na tela. Para isso dispõe de recursos como a interface Ivanita.

O equipamento mais bem servido de software para processamento de texto é o IBM PC e os compatíveis, pois seu sistema de processamento de 16 bits e sua grande capacidade de memória permitem a utilização de softwares poderosos e sofisticados, como o WordStar, o Word, o Multimate e o Symphony.

Este último faz parte de uma nova tendência na área, a dos softwares integrados, que reúnem os recursos de planilha eletrônica, gráficos, processador de textos, banco de dados e, em alguns casos, telecomunicações. Assim, com a tela dividida em "janelas", pode-se fazer com que uma carta incorpore dados de um gráfico, de um banco de dados ou de uma planilha financeira.

Alguns desses programas são direcionados por menus, isto é, a tela mostra permanentemente uma lista de funções mais utilizadas, indicadas por palavras ou, em alguns casos, por figuras.

Outros oferecem comandos fáceis de memorizar, com teclas associadas à letra inicial da palavra de comando; outros, ainda, têm os comandos principais associados às teclas de função, que são teclas especiais encontráveis nos computadores de maiores recursos, programáveis para substituir toda uma série de comandos. Convém que o programa mostre mensagens de erro esclarecedoras ou disponha de telas explicativas de auxílio ao usuário.

Criação e organização

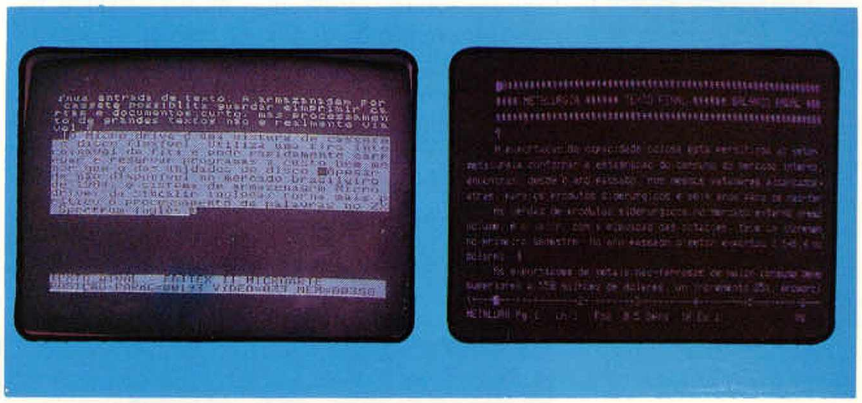
Quem lida com aplicações comerciais conta com recursos específicos. Eles permitem, por exemplo, elaborar textos complexos, repetitivos e que exigem precisão, como contratos, fórmulas jurídicas e descrição de produtos. No caso, esses textos são compostos pela junção de vários parágrafos padronizados.

Pode-se ainda enviar a mesma carta a grande número de pessoas, mudando apenas as informações referentes ao destinatário. Tais cartas são produzidas com os softwares de Mala Direta (Mailmerge). Oferecidos por diversos processadores, esses recursos permitem imprimir muitas cópias da mesma carta, cada qual associada a um dos registros de arquivo de nomes e endereços. Da mesma forma, é possível redigir uma carta padrão na qual variam apenas certos dados.

Os programas mais sofisticados já possibilitam escrever cartas condicionais, nas quais alguns segmentos do texto dependem de variáveis encontradas no arquivo de endereços. Assim, por exemplo, conforme o sexo do destinatário, a carta dirá "Prezado Senhor" ou "Prezada Senhora", e uma carta de cobrança pode adaptar o prazo limite e o valor do pagamento aos dados referentes ao cliente.

Há programas que manipulam arquivos com grande rapidez; outros gravam continuamente o texto no disquete, evitando que se percam informações no caso de faltar energia elétrica ou falhar o equipamento.

Alguns programas dispõem de um buffer (reservatório) para armazenamento, com o que libertam o micro para processamento de um texto enquanto outro texto é transmitido do buffer para a impressora. Há também programas que enfileiram "ordens de serviço" para que vários textos sejam impressos em seqüência.



Ferramenta poderosa
O Editex, desenvolvido no Brasil, permite verificar a ortografia e a acentuação. O programa dispõe de 132 regras armazenadas.
Auxiliares eficientes em diversas atividades, os editores de texto ajudam agora o trabalho jornalístico, possibilitando produzir e enviar à redação em poucos minutos um texto final como o da ilustração à direita.

Os equipamentos de 16 bits já dispõem de novos tipos de software, à venda no mercado internacional, que vão além dos aspectos mecânicos da escrita e revisão dos textos, oferecendo recursos de auxílio à própria criação e organização das idéias. O THOR (THought ORganizer), armazena informações relativas a várias áreas ou projetos diferentes, classifica-as e indexa-as de acordo com diferentes critérios. Depois são pesquisadas, agrupadas e acessadas as informações selecionadas, na tela ou na impressora. Sem ser propriamente um banco de dados, o THOR oferece excelentes recursos de armazenamento e referência cruzada de textos.

Outra poderosa ferramenta é o WANDAH (Writing Aid AND Author's Helper), que auxilia a elaboração de um texto desde o planejamento inicial e a divisão do assunto em itens e subitens até a revisão final de ortografia, pontuação, concordância e mesmo estilo. Adverte o autor, por exemplo, quanto à repetição excessiva ou ao uso indevido de termos.

Com isso, chega-se ao limiar da era da inteligência artificial aplicada à linguagem escrita. Reunindo um grande corpo de conhecimentos dos processos complexos e muitas vezes irracionais da linguagem humana, os processadores de texto serão auxiliares cada vez mais eficientes de todos aqueles que trabalham com palavras.

ESCOLHA DE ARQUIVOS

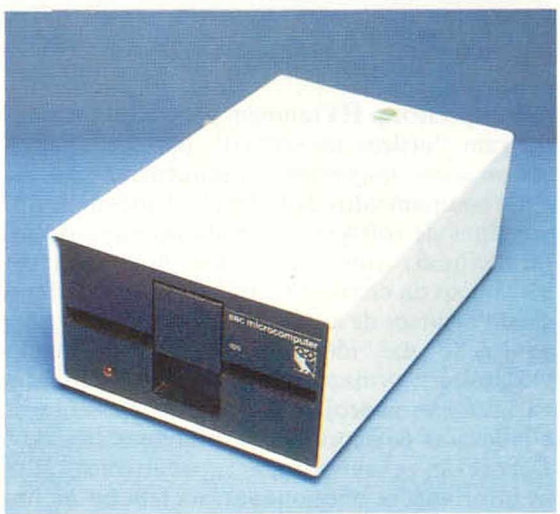
O baixo preço de cartuchos e fitas cassete deu-lhes a preferência dos programadores quando se trata de armazenar dados. Mas os modernos discos flexíveis já se tornaram acessíveis a todo tipo de usuário.

CONTATO PERIGOSO

Nunca deixe o disquete próximo a ímãs. Dispositivos como os telefones possuem eletroímãs e os alto-falantes dos sistemas de som são dotados de potentes ímãs, capazes de alterar os dados gravados.

Embora sejam ferramentas muito versáteis para o tratamento de dados, os microcomputadores de nada servem se não dispuserem de um meio de reter a informação. Nos sistemas de armazenamento de dados incluem-se os cartuchos de ROM, as fitas cassete comuns e os discos magnéticos flexíveis, ou disquetes.

Preço, capacidade de armazenamento e facilidade de acesso aos dados são fatores que determinam a opção por um ou outro tipo de memória.



Unidade de disquete BBC
Para acoplá-la ao micro BBC é preciso instalar o DOS ROM no próprio computador. Esse procedimento é desnecessário nas unidades de disco "inteligentes", as quais já vêm equipadas com um circuito integrado DOS.

Cartucho

O armazenamento em cartucho apresenta poucas características favoráveis ao programador. Na maior parte dos cartuchos, a memória é programável apenas para leitura (PROM, Programmable Read Only Memory). Esse tipo funciona somente como meio de introdução de dados no computador, em geral sob a forma de jogos, escritos no extenso e complexo código de máquina. A PROM funciona também como recurso suplementar — por exemplo, extensões para o BASIC. Mas já foram desenvolvidos cartuchos com PROM apagáveis eletricamente (EEPROM, Electrically Erasable Programmable Read Only Memory), que podem ser gravadas e lidas do mesmo modo que a RAM interna. Ao contrário desta, contudo, a EEPROM não é volátil: a informação permanece registrada quando se removem os cartuchos ou quando se desliga o computador.

Há outros cartuchos de características semelhantes, disponíveis apenas para alguns computadores. Operando com circuitos de RAM que utilizam semicondutor complementar de óxido metálico (CMOS, Complementary Metal Oxide Semiconductor), tais cartuchos consomem pouca energia e conseguem manter a informação armazenada graças a uma bateria instalada em seu interior.

O principal argumento contra o armazenamento de dados em EEPROM e RAM CMOS é o alto preço desses tipos de memória. A formação de uma biblioteca de tais cartuchos, mesmo modesta, é quase tão dispendiosa quanto a aquisição de uma unidade de discos flexíveis.

MEMÓRIAS DE CURTO E DE LONGO PRAZO
A memória de acesso aleatório (RAM, Random Access Memory) armazena as informações que podem ser alteradas pelo usuário. É, portanto, uma memória de curto prazo. A memória somente de leitura (ROM, Read Only Memory) contém as informações permanentes, que não podem ser alteradas. Trata-se de uma memória de longo prazo.

Fitas cassete

Baratas e facilmente disponíveis, as fitas cassete funcionam nos gravadores portáteis, que podem ser conectados a quase todos os microcomputadores. Por isso, continuam a ser o meio mais empregado para o armazenamento de programas e dados.

O armazenamento se faz em arquivos seqüenciais, sob a forma binária: os zeros e os uns correspondem a tons diferentes. A gravação do nome do arquivo, de seu endereço na memória e de seu conteúdo é realizada bit a bit em grupos de 1 byte (8 bits) e segmentos de 256 bytes. Alguns computadores contam com um recurso denominado "soma de verificação", que permite verificar eventuais erros de gravação.

Os comandos básicos em BASIC nesse sistema de armazenamento, são SAVE, para gravar os arquivos na fita, e LOAD, que permite o acesso aos dados e a leitura deles.

Em alguns sistemas há comandos adicionais para várias funções específicas, como gerar um catálogo com os nomes dos arquivos armazenados ou formatar a gravação e leitura de dados de diferentes tipos.

O baixo custo e a facilidade de compreensão dos comandos são vantagens pequenas quando comparadas aos vários inconvenientes desse tipo de armazenamento.

Na grande maioria dos casos, o usuário se vê obrigado a operar manualmente os controles para o acesso aos dados. É necessário, assim, um ajuste preciso do volume do gravador e muito cuidado no acionamento das teclas, o que torna a operação um tanto demorada.

Outro inconveniente diz respeito ao modo seqüencial do armazenamento de dados em fitas.

Se quiser recuperar um arquivo específico, o operador deverá estar atento ao contador de giros do gravador para voltar a fita na posição exata (um pouco antes do início do arquivo). Do contrário, o computador será obrigado a pesquisar toda a fita. E dentro de cada arquivo é impossível recuperar os itens de imediato, sem precisar processá-lo por inteiro.

A todas as desvantagens relacionadas, soma-se a baixa velocidade de gravação e leitura do sistema: de 300 a 1.200 bits por segundo. Um programa curto, de 5 Kbytes, por exemplo, pode levar de 1 a 3 minutos para ser carregado ou gravado. A baixa velocidade do sistema dificulta também a obtenção das valiosas cópias de reserva (backup) dos programas.

Desgaste da fita, provocando imprevisíveis alterações nos dados, e possibilidade de rupturas devidas ao sistema de transporte da fita completam a lista das desvantagens desse tipo de armazenamento.

Disco flexível

Comparados aos sistemas de armazenamento em cartucho ou cassete, os discos (ou disquetes) flexíveis apresentam poucos inconvenientes. As unidades de disquete (drives) são aparelhos complexos e delicados — o que torna seu preço consideravelmente elevado. Os disquetes também são caros, mas para o usuário constituem um meio confiável e versátil de armazenar grande quantidade de dados. A velocidade de gravação e leitura em discos flexíveis chega a ser até duzentas vezes maior do que a das fitas cassete.

Todas as unidades dispõem de algum tipo de sistema operacional de disco (DOS, Disk Operating System), que contém uma rotina responsável pela distribuição dos dados pelas trilhas do disco. Em geral, existem de 35 a 80 trilhas por lado, cada uma dividida em número variável de setores. As trilhas próximas ao centro do disco são menores e possuem menos setores do que as externas. Cada setor, por sua vez, abrange um bloco de dados, em geral de 256 bytes.

O DOS registra as posições em que estão gravados todos os dados armazenados no disco. Isso é possível graças à existência de um mapa de disponibilidade de blocos (BAM, Block Availability Map) — gravado no próprio disco ou na memória principal do computador — e de um catálogo (ou diretório). O BAM mantém um registro dos blocos em uso e dos que estão livres para novo armazenamento; o catálogo é uma lista dos nomes e tipos de arquivo e da localização de cada um por trilha e por setor. Costuma-se gravá-lo na trilha central e também carregá-lo na memória do computador, para funcionar como referência. Após consultar o BAM, o DOS direciona o movimento da cabeça de leitura/gravação para que se efetuem as operações de armazenamento e recuperação de dados.

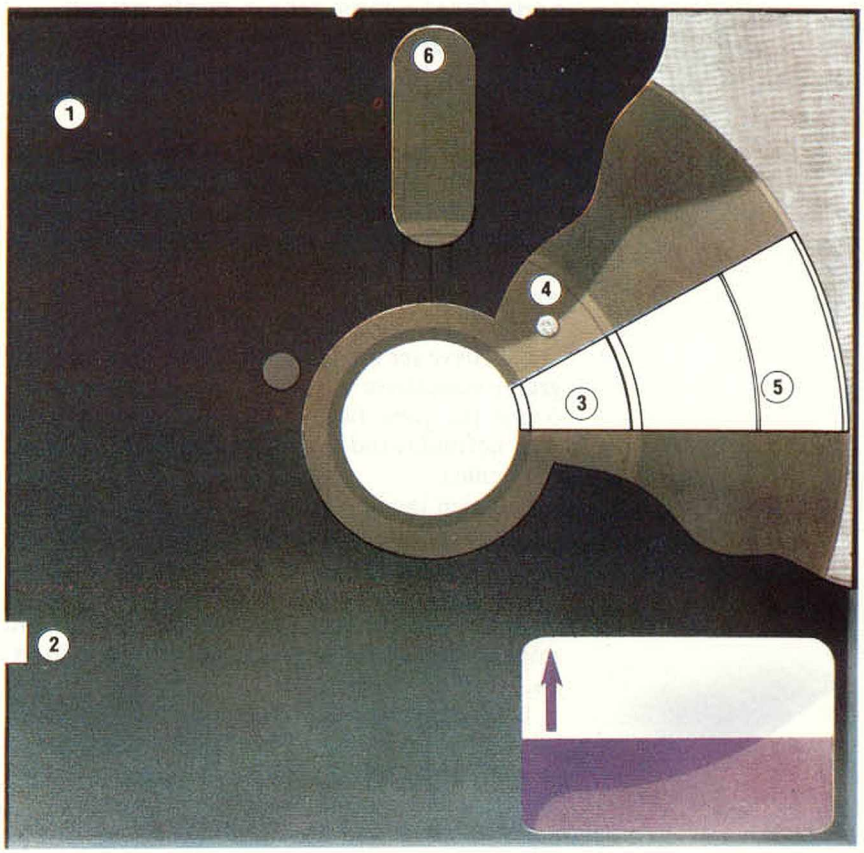
A alocação dos dados em trilhas e setores e o posicionamento exato da cabeça de leitura/gravação tornam possível o acesso direto aos arquivos. Se necessário, os bits podem ser gravados ou lidos em grupos de até 1 byte. De modo geral, as diferenças entre as unidades de disquete são pequenas: limitam-se à quantidade de dados — de 100 a 400 Kbytes — que pode ser armazenada; à velocidade de transferência dos dados; e às maneiras de se controlar o registro e a leitura dos dados por meio do DOS.

Há três métodos principais para se instalar um DOS. O mais eficiente é implantá-lo, em forma de ROM, na unidade de disco, sob o controle do microprocessador da própria unidade, associado à RAM. Essas unidades de disco são chamadas "inteligentes", pois, ao receberem uma instrução da unidade central de processamento (CPU), podem executar independentemente as complexas rotinas de operação dos discos. Assim, o processador do microcomputador fica liberado para continuar executando o programa principal.

O método mais comum, no entanto, é o que transfere o DOS, armazenado no disco, para a RAM do computador. Isso pode ser feito automaticamente ou por meio de um comando manual, quando se aciona o computador.

O último método consiste na inclusão de um tipo de DOS no próprio sistema operacional do computador (gravado numa EPROM). As rotinas para a operação dos discos incluem os comandos do tipo SAVE e LOAD, um comando CAT (catalog) para o diretório, um comando para a formatação do disquete (ou do cartucho) e vários comandos para criação, manipulação e eliminação de arquivos seqüenciais ou de acesso direto.

Gravação em disquete
Os disquetes são feitos de substância plástica resistente, coberta por uma camada magnetizável de óxido metálico. Protegido por um envelope plástico quadrado, o disco é girado por meio de seu eixo. Uma abertura no envelope (6) permite o acesso da cabeça de leitura/gravação à superfície de registro.



1 ENVELOPE PROTETOR
2 ENTALHE DE PROTEÇÃO CONTRA GRAVAÇÃO
3 SETOR
4 ORIFÍCIO DE ALINHAMENTO
5 TRILHA
6 ABERTURA DE ACESSO À CABEÇA DE LEITURA/GRAVAÇÃO

LIÇÕES E DESAFIOS

Uma série de softwares educativos exemplificam o que a informática pode fazer nessa área e quais os erros a serem evitados pelo programador voltado para a criança.

Os jogos aqui examinados têm objetivos educacionais que variam desde o simples reconhecimento de cores e formas até o desenvolvimento da criatividade infantil, passando pelas habilidades básicas de leitura e manipulação de números. Todos eles explicam ao jovem usuário, de modo direto, o que fazer.

Esta deve ser uma prioridade em qualquer programa educativo: a criança precisa compreender o que lhe cabe fazer, recebendo recompensas bem definidas toda vez que domina determinada técnica.

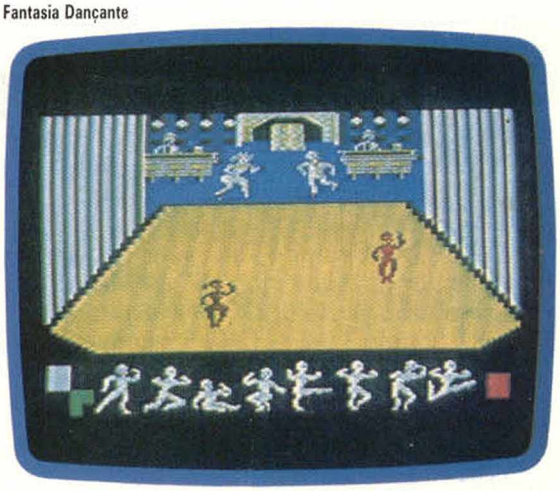
Convém igualmente que o programa seja fácil de usar. Não haveria utilidade num software que se destinasse a ensinar a criança a ler e começasse com uma lista de instruções operacionais. Os melhores programas apresentam um mínimo desse tipo de instrução inútil e, não raro, complicadora.

Um bom programa educativo é o que mantém o interesse da criança. Por mais valioso que seja seu objetivo, não o atingirá se estiver acima da capacidade do usuário ou se for repetitivo.

Outra exigência fundamental do programa educativo: que ele de fato ensine o que se propõe ensinar. Isso parece evidente, mas muitas vezes os fabricantes de software esquecem o objetivo educacional ao favorecer o valor de entretenimento.

Os pacotes aqui apresentados, produzidos pelas companhias americanas Spinnaker e Fisher-Price, são encontrados em cartuchos ou fitas cassete no mercado internacional. A criança pequena aprende com facilidade a colocar um cartucho no computador e ligá-lo. Já a fita cassete exige a presença de um adulto para ajudar a carregar o programa.

Fantasia Dançante

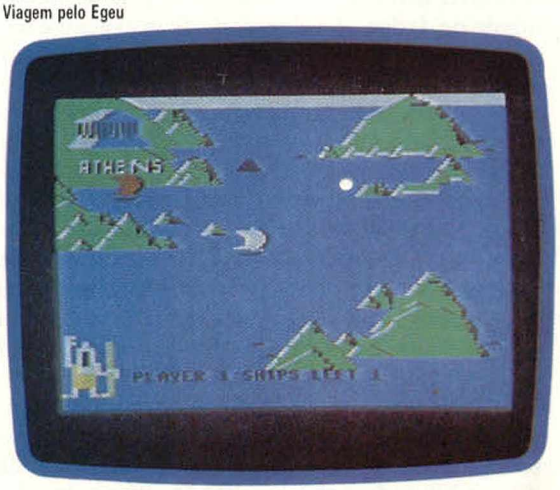


Danças e viagens

Dance Fantasy (Fantasia Dançante) tem como público-alvo crianças de quatro a oito anos. A imagem na tela mostra um palco no qual estão duas figuras em pé. A criança é convidada a coreografar uma dança para elas. Antes de começar, ela especifica o sexo dos dançarinos: um menino e uma menina, dois meninos, duas meninas.

Na parte inferior da tela aparecem figuras em várias poses: cada uma representa determinada seqüência de dança (salto, pirueta etc.). Ao mover cada dançarino sobre uma dessas figuras, usando o joystick, a criança escolhe a seqüência específica. Depois posiciona as figuras no palco

Viagem pelo Egeu



e, pressionando o botão de disparo, faz com que elas dancem.

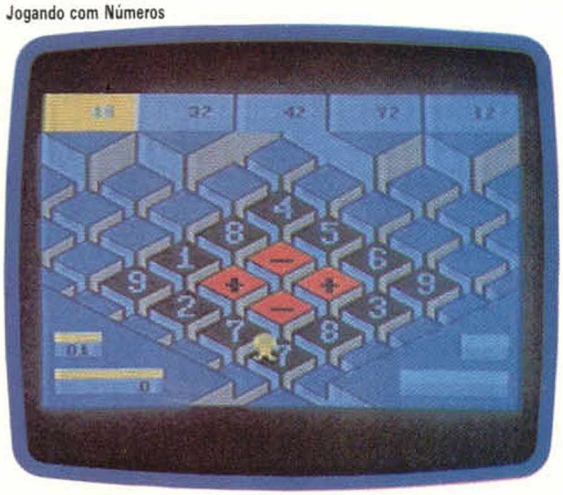
Assim, a dança é coreografada pela seleção de uma série de movimentos e de sua execução em pontos diferentes do palco. O programa fornece os movimentos de ligação. Completada a dança, a criança pode gravá-la e visualizar o efeito final.

A força de Dance Fantasy reside em sua imaginosa analogia com um programa de computador: a criança pode montar sua própria dança *(programa)* usando uma série de seqüências básicas (um conjunto de *rotinas)*. Depois *grava* o resultado em fita cassete e o *carrega* — assim, sem dificuldade, se familiariza com esses conceitos.

Aegean Voyage (Viagem pelo Egeu) é para crianças um pouco maiores. Usa personagens e cenários da mitologia grega como elementos de um jogo de aventuras. O objetivo: fazer um navio viajar de Atenas até várias ilhas do mar Egeu, evitando recifes e tempestades. Quando o jogador chega à segurança de um porto, surgem na tela o nome da ilha e uma mensagem cifrada. Ele então decide se a ilha deve ser explorada. Se a decisão é correta, recebe como recompensa um tesouro (por exemplo, o escudo de Aquiles); se é errada, recebe punição: o afundamento do navio por uma criatura como a Górgona.

O estudioso dos clássicos notará que o jogo apresenta alguns fatos errados: o Minotauro, por exemplo, aparece tanto em Delos como em Cre-

Jogando com Números



ta. Além disso, o jogo não explica o significado de nomes e lugares. É mínima a probabilidade de a criança obter conhecimentos razoáveis da mitologia grega com esse cassete.

Ele não mantém a atenção da criança por muito tempo, pois as formas e os gráficos são repetitivos e pouco interessantes.

Números, nomes e formas

Projetado para desenvolver a habilidade aritmética de garotos de oito a doze anos, Number Tumblers (Jogando com Números) proporciona a velocidade e a emoção de um fliperama. Apresenta na parte de cima da tela uma série de números, e o jogador deve ordenar os símbolos aritméticos e numéricos nos lados de um conjunto de dados, para criar uma expressão matemática que corresponda a um dos números.

É um jogo rápido e divertido, com gráficos atraentes e bem desenhados. Um incentivo real para a criança desenvolver a habilidade em aritmética.

Kindercomp (Computador Infantil) destina-se a meninos de três a oito anos. O pacote — com uma série de diferentes exercícios — visa a familiarizar a criança com o uso do computador e a desenvolver suas aptidões artísticas. Mas dá a impressão de se limitar a uma série de truques com o computador — o tipo de coisa que a maioria dos programadores faz ao aprender BASIC e testar as capacidades da máquina.

Computador Infantil

Numa das opções, Nomes, o programa convida o jogador a fornecer um nome ou uma sentença de até quinze caracteres de comprimento. As letras aparecem em cores e tamanhos variados por toda a tela. O efeito visual, para a criança não acostumada com os gráficos do computador, parece fabuloso. Contudo, o programa tem pouco valor educativo, pois qualquer grupo de letras proporciona o mesmo efeito.

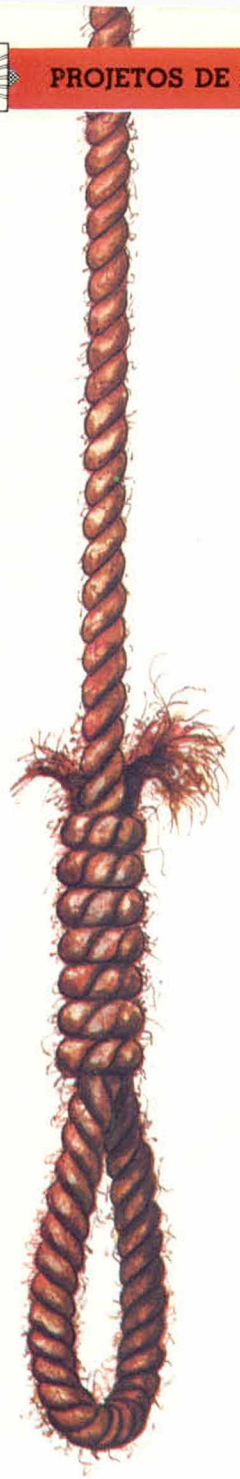
O Kindercomp pode não manter o interesse da criança por muito tempo — seus truques são divertidos mas limitados. É o tipo de pacote com que os garotos se entretêm por uns três dias — e depois nunca mais usam.

Alf In The Color Caves (Alf nas Cavernas Coloridas) apresenta um divertido personagem — Alf — que escorrega por uma série de tubos coloridos e de várias formas até cair numa sala no fundo das cavernas. Usando um joystick ou o teclado, o jogador guia o personagem pelas cavernas. Se o conduzir com segurança, passando por ameaçadores pares de olhos que se movem rapidamente, será recompensado com o espetáculo de uma graciosa dança de Alf. Depois, o personagem é sugado por um tubo até o chão, para reiniciar a descida.

Esse é um programa educativo funcional, adequado a crianças pequenas. Pode servir de exemplo aos produtores dos softwares destinados ao desenvolvimento de habilidades infantis.

Alf nas Cavernas Coloridas

A PRÓXIMA LETRA



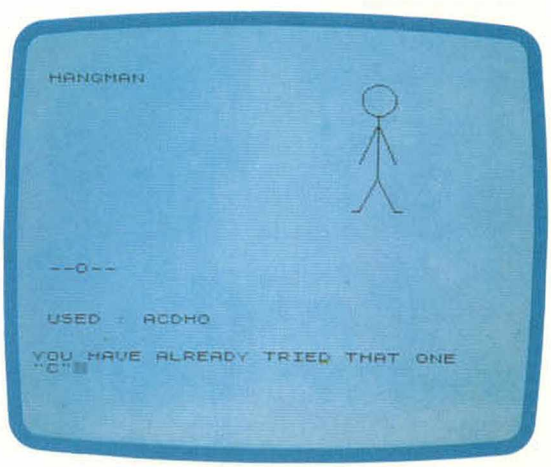
A programação desse tradicional jogo de palavras constitui também um bom exercício para a manipulação de strings. Aqui, as três versões de sua estrutura básica: para as linhas Sinclair, Apple e TRS-80.

Poucos jogos são tão conhecidos em todo o mundo quanto o de forca. Na versão eletrônica desse passatempo, o jogador tem por objetivo adivinhar, letra por letra, uma palavra escolhida aleatoriamente pelo micro entre onze determinadas pelo programador. Cada palavra tem no máximo dez letras, e os espaços correspondentes estão indicados por traços horizontais na tela. Todas as letras tentadas pelo jogador permanecem na tela, lembrando que já foram usadas. Quando o jogador acerta uma letra, ela é colocada no lugar a que pertence na palavra. A cada tentativa errada, o micro desenha uma das dez partes que formam a figura de um homem. Se o jogador cometer dez erros, o boneco é enforcado: o jogador perde a partida e o aparelho seleciona uma nova palavra.

Na versão para a linha Sinclair, as palavras não estão armazenadas no programa; elas são digitadas pelo programador, no início de cada jogo. Para iniciar, digite RUN [NEW LINE] e uma mensagem solicitará onze palavras para que uma seja sorteada.

É fácil, nas três versões, modificar o número de letras por palavra, assim como o número total de palavras.

Como um desafio extra, você pode melhorar seu programa, acrescentando, por exemplo, uma melodia de congratulações ou de pêsames; ou tentar fazer o enforcado balançar, ao fim do jogo.



Um momento do jogo
Na tela, a evolução do programa, mostrando as tentativas já feitas (A, C, D, H, O), a letra correta (O) entre os traços e as partes do corpo desenhadas, além do aviso do computador de que já foi tentada a letra C.

TRS-80

```
10  REM ****************************
11  REM *      JOGO DA FORCA       *
12  REM ****************************
20  CLEAR 1000:DIM P$(12)
30  REM ***  INICIALIZAR ***
40  D$="----------"
50  FOR I=1 TO 11
60  READ P$(I)
70  NEXT I
80  REM ***  NOVA PALAVRA  ***
90  CLS
100 PRINT @30,"FORCA"
110 P$=P$(RND(11))
120 C=0:P=0:D=0
130 G$=""
140 PRINT @768,"LETRAS: ";
150 L=LEN(P$)
160 PRINT @640,LEFT$(D$,L)
170 REM ***  INPUT LETRA  ***
180 PRINT @896,"ESCOLHA A LETRA: ";STRING$(45,32):PRINT
    @913,"";:INPUT L$
190 REM ***  VERIFICA LETRA  ***
200 IF L$="" THEN GOTO 180
210 IF LEN(L$)>1 THEN GOTO 180
220 F=0
230 FOR I=1 TO LEN(G$)
240 IF L$<>MID$(G$,I,1) THEN GOTO 270
250 PRINT @913,"VOCE JA USOU ESTA LETRA";:FOR X=1 TO
    1000:NEXT X
260 F=1
270 NEXT I
280 IF F=1 THEN GOTO 170
290 G$=G$+L$
300 PRINT @776,G$;
310 REM *** TESTE DA LETRA NA PALAVRA ***
320 F=0
330 FOR I=1 TO L
340 IF L$=MID$(P$,I,1) THEN GOSUB 500
350 NEXT I
360 IF F<>1 THEN GOSUB 800
370 IF D=1 THEN GOSUB 700:GOTO 80
380 IF C=L THEN GOSUB 600:GOTO 80
390 GOTO 170
400 END
500 REM *** ACERTOU A LETRA ***
510 PRINT @639+I,L$;
520 C=C+1
530 F=1
540 RETURN
600 REM *** ACERTOU A PALAVRA ***
610 PRINT @896,"PARABENS!!!          ACERTOU A PALAVRA";
    STRING$(27,32);
620 FOR I=1 TO 3000:NEXT I
630 RETURN
700 REM *** PERDEU O JOGO ***
710 PRINT @896,"ENFORCOU-SE! A PALAVRA ERA  ";P$;
720 FOR I = 1 TO 2000:NEXT I
730 RETURN
800 REM *** ERROU A LETRA ***
810 P=P+1
820 ON P GOTO 830,840,850,870,880,890,900,910,920,930,
    940
830 REM *** DESENHA CABECA ***
831 PRINT @50,CHR$(151);CHR$(131);CHR$(131);CHR$(131);
    CHR$(171);:PRINT@114,CHR$(149);"   ";CHR$(170);:
    PRINT @178,CHR$(141);CHR$(140);CHR$(140);CHR$(140);
    CHR$(142);:RETURN
840 REM *** DESENHA PESCOCO ***
841 PRINT @180,CHR$(188);:PRINT @244,CHR$(131);:RETURN
850 REM *** DESENHA CORPO ***
851 PRINT @241,CHR$(151);CHR$(131);CHR$(131);CHR$(131);
    CHR$(131);CHR$(131);CHR$(171);:PRINT@305,CHR$(149);
    STRING$(5,32);CHR$(170);:PRINT @369,CHR$(149);
    STRING$(5,32);CHR$(170);
860 PRINT @433,CHR$(149);STRING$(5,32);CHR$(170);:
    PRINT @497,CHR$(131);CHR$(131);CHR$(131);
    CHR$(131);CHR$(131);CHR$(131);CHR$(131);: RETURN
870 REM *** DESENHA BRACO ESQUERDO ***
871 PRINT @237,CHR$(131);CHR$(131);CHR$(131);CHR$(131);
    RETURN
880 REM *** DESENHA BRACO DIREITO ***
881 PRINT @248,CHR$(131);CHR$(131);CHR$(131);CHR$(131);
    RETURN
890 REM *** DESENHA PERNA ESQUERDA ***
891 PRINT @497,CHR$(171);:PRINT @561,CHR$(170);:PRINT
    @625,CHR$(170);:RETURN
900 REM *** DESENHA PERNA DIREITA ***
901 PRINT @503,CHR$(151);:PRINT @567,CHR$(149);:PRINT
    @631,CHR$(149);:RETURN
910 REM *** DESENHA PE' ESQUERDO ***
911 PRINT @624,CHR$(176);CHR$(186);:RETURN
920 REM *** DESENHA PE' DIREITO ***
921 PRINT @631,CHR$(181);CHR$(176);:RETURN
930 REM *** DESENHA MAO ESQUERDA ***
931 PRINT @237,CHR$(151);:RETURN
940 REM *** DESENHA MAO DIREITA ***
941 PRINT @251,CHR$(171);:D=1:RETURN
950 REM *** PALAVRAS SECRETAS ***
960 DATA "TIGELA","QUERUBIM","EXORCISTA","XENOFOBIA",
    "ARCEBISPO"
970 DATA  "ISTMO","ESTUPIDO","ENFORCADO","ESTRANHO",
    "REVISTA","MASCULO"
```

Sinclair

```
  5 REM **********************
 10 REM *   JOGO DA FORCA    *
 15 REM **********************
 20 REM ***  ENTRADA PARA 11 PALAVRAS  ***
 30 LET N=11
 40 DIM X$(N,10)
 45 DIM Y(N)
 50 LET D$="----------"
 55 PRINT "DIGITE A PALAVRA"
 60 FOR I=1 TO N
 65 PRINT AT 2,8;I
 70 INPUT W$
 75 LET Y(I)=LEN W$
 80 LET X$(I)=W$
 90 NEXT I
100 CLS
110 PRINT AT 1,1; "JOGO DA FORCA"
120 LET I=1+INT (RND*11)
130 LET W$=X$(I)
140 LET C=0
150 LET W=0
160 LET G$=""
170 LET D=0
180 PRINT AT 20,1; "USADAS : "
185 LET L=Y(I)
190 PRINT AT 16,1;D$(1 TO L)
200 REM ***  ENTRADA DA LETRA  ***
220 INPUT L$
230 IF L$ = " " THEN GOTO 220
240 IF LEN L$>1 THEN GOTO 220
250 LET F=0
270 FOR I=1 TO LEN G$
280 IF L$<>G$(I) THEN GOTO 330
290 PRINT AT 21,1; "ESTA VOCE JA TENTOU..."
310 PAUSE 100
315 PRINT AT 21,1; "                      "
320 LET F=1
330 NEXT I
340 IF F=1 THEN GOTO 220
350 LET G$=G$+L$
360 PRINT AT 20,10;G$
370 REM ***  TESTE DA LETRA NA PALAVRA  ***
380 LET F=0
390 FOR I=1 TO L
400 IF L$=W$(I) THEN GOSUB 480
410 NEXT I
420 IF F<>1 THEN GOSUB 670
430 IF D=1 THEN GOSUB 590
435 IF D=1 THEN GOTO 100
440 IF C=L THEN GOSUB 530
445 IF C=L GOTO 100
450 GOTO 220
460 STOP
470 REM ***  A LETRA COMBINA  ***
480 PRINT AT 16,11;L$
490 LET C=C+1
500 LET F=1
520 RETURN
530 REM ***  SUCESSO...  ***
540 PRINT AT 10,11;"***VOCE VENCEU***"
550 PAUSE 100
560 PRINT AT 10,11;"                "
570 RETURN
580 REM ***  FRACASSO...  ***
590 PRINT AT 10,11; "***VOCE PERDEU***"
600 PAUSE 100
610 CLS
630 PRINT AT 10,3; "A PALAVRA ERA : ";W$
640 PAUSE 100
650 RETURN
660 REM ***  NAO E A LETRA  ***
670 LET W=W+1
680 IF W=1 THEN GOTO 700
681 IF W=2 THEN GOTO 750
682 IF W=3 THEN GOTO 800
683 IF W=4 THEN GOTO 850
684 IF W=5 THEN GOTO 900
685 IF W=6 THEN GOTO 950
686 IF W=7 THEN GOTO 1000
687 IF W=8 THEN GOTO 1050
688 IF W=9 THEN GOTO 1100
689 IF W=10 THEN GOTO 1150
700 REM ***  CORDA ***
720 PRINT AT 2,24; "*"
730 PRINT AT 3,24; "*"
745 RETURN
750 REM ***  CABECA  ***
755 PRINT AT 4,22; "*****"
760 PRINT AT 5,22; "*   *"
770 PRINT AT 6,22; "*   *"
780 PRINT AT 7,22; "*   *"
785 PRINT AT 8,22; "*****"
790 RETURN
800 REM ***  OLHOS E BOCA  ***
810 PRINT AT 5,23; "O O"
820 PRINT AT 7,24; "-"
830 RETURN
850 REM ***  PESCOCO  ***
860 PRINT AT 9,24; "*"
870 PRINT AT 10,24; "*"
880 RETURN
900 REM ***  PARTE DO CORPO  ***
910 PRINT AT 11,22; "*****"
920 PRINT AT 12,22; "*   *"
930 PRINT AT 13,22; "*   *"
940 RETURN
950 REM ***  RESTO DO CORPO  ***
960 PRINT AT 14,22; "*   *"
970 PRINT AT 15,22; "*   *"
980 PRINT AT 16,22; "*****"
990 RETURN
1000 REM ***  BRACO  ***
1010 PRINT AT 11,19; "***"
1020 RETURN
1050 REM ***  OUTRO BRACO  ***
1060 PRINT AT 11,27; "***"
1070 RETURN
1100 REM ***  PERNA  ***
1110 PRINT AT 17,22; "*"
1120 PRINT AT 18,22; "*"
1130 PRINT AT 19,22 "*
1140 RETURN
1150 REM ***  OUTRA PERNA  ***
1160 PRINT AT 17,26; "*"
1170 PRINT AT 18,26; "*"
1180 PRINT AT 19,26; "*"
1190 LET D=1
1200 RETURN
```

Apple

```
]LIST

100  REM   ********************
110  REM   *  JOGO DA FORCA  *
120  REM   ********************
125  REM
130  TEXT
135  CLEAR : DIM PS(20)
140  GOSUB 490
145  VTAB 21: HTAB 1: PRINT "TECLE UMA LETRA [ ]"
150  VTAB 21: HTAB 18: GET TS:N = 0:BS = ""
155  LET SOM =  PEEK ( - 16322): FOR I = 1 TO 26
160  IF TS =  MIDS (AS,I,1) THEN N = I:I = 26: GOTO 170
165  IF I = 26 THEN  GOSUB 595
170  NEXT
175  FOR I = 1 TO 26
180  IF I = N THEN BS = BS + " ": GOTO 190
185  LET BS = BS +  MIDS (AS,I,1)
190  NEXT
195  VTAB 23: HTAB 1:AS = BS: PRINT AS
200  REM  *** EXISTE ESTA LETRA ? ***
205  LET C = 0: FOR I = 1 TO  LEN (PS(P))
210  IF TS =  MIDS (PS(P),I,1) THEN  GOSUB 615
215  NEXT
220  LET TA = TA + C: IF TA =  LEN (PS(P)) THEN  GOTO 435
225  IF C =  > 1 THEN  GOTO 145
230  LET BR = BR + 1
235  REM *** O QUE SERA DESENHADO ***
240  ON BR GOTO 245,270,285,305,320,335,350,365
245  REM *** DESENHA A CABECA ***
250  PLOT 19,12: PLOT 20,12: PLOT 21,12
255  PLOT 19,13: PLOT 21,13: PLOT 19,14
260  PLOT 20,14: PLOT 21,14
265  GOTO 145
270  REM *** DESENHA O PESCOCO ***
275  PLOT 20,15
280  GOTO 145
285  REM *** DESENHA O CORPO ***
290  HLIN 18,22 AT 16: HLIN 18,22 AT 26
295  VLIN 16,26 AT 18: VLIN 16,26 AT 22
300  GOTO 145
305  REM *** DESENHA O BRACO ESQUERDO ***
310  HLIN 14,17 AT 17
315  GOTO 145
320  REM ***DESENHA O BRACO DIREITO ***
325  HLIN 23,26 AT 17
330  GOTO 145
335  REM *** DESENHA A PERNA ESQUERDA ***
340  VLIN 27,36 AT 19
345  GOTO 145
350  REM *** DESENHA A PERNA DIREITA ***
355  VLIN 27,36 AT 21
360  GOTO 145
365  REM *** ENFORCA E ESCREVE HA! ***
370  COLOR= 3: VLIN 5,13 AT 24: VLIN 5,13 AT 27
375  VLIN 5,13 AT 29: VLIN 5,13 AT 32: VLIN 5,10 AT 34
380  PLOT 25,9: PLOT 26,9: PLOT 30,5: PLOT 31,5:
385  PLOT 30,9: PLOT 31,9: PLOT 34,13
390  PRINT  CHRS (7)
395  VTAB 21: HTAB 1: PRINT  SPC( 255): VTAB 21
400  HTAB 1: PRINT "VOCE PERDEU"
405  PRINT "A PALAVRA ERA ";PS(P)
410  PRINT "VOCE QUER JOGAR OUTRA VEZ ? (S/N)"
415  VTAB 21: HTAB 39: GET QWS
420  IF QWS <  > "S" AND QWS <  > "N" THEN  GOTO 415
425  IF QWS = "S" THEN  RUN
430  TEXT : HOME : END
435  REM  *** VOCE ACERTOU ***
440  PRINT  CHRS (7)
445  VTAB 21: HTAB 1: PRINT  SPC( 255): VTAB 21
450  HTAB 1: PRINT "VOCE ACERTOU"
455  PRINT "A PALAVRA E ";PS(P)
460  PRINT "VOCE QUER JOGAR OUTRA VEZ ? (S/N)"
465  VTAB 21: HTAB 39: GET QWS
470  IF QWS <  > "S" AND QWS <  > "N" THEN  GOTO 465
475  IF QWS = "S" THEN  RUN
480  TEXT : HOME : END
485  REM   *** QUAL A PALAVRA ***
490  LET P =  INT ( RND (1) * 11)
495  LET PS(0) = "MACACO":PS(1) = "ELEFANTE"
500  LET PS(2) = "DICIONARIO":PS(3) = "COMPUTADOR"
505  LET PS(4) = "IMAGINACAO"
510  LET PS(5) = "TELEFONE":PS(6) = "GEOGRAFIA"
515  LET PS(7) = "AVENIDA":PS(8) = "ARGENTINA"
520  LET PS(9) = "MARGARIDA":PS(10) = "TRANSITO"
525  TEXT : HOME : GR : POKE 34,24
530  VTAB 22:AS = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
532  IF  LEN (PS(P)) > 10 THEN  STOP
535  PRINT ">";: FOR I = 1 TO  LEN (PS(P))
540  PRINT "-";: NEXT : PRINT "<": PRINT AS
545  COLOR= 3
550  VLIN 10,39 AT 5
555  HLIN 5,20 AT 9
560  PLOT 6,16: PLOT 7,15: PLOT 8,14: PLOT 9,13
565  PLOT 10,12: PLOT 11,11: PLOT 12,10
570  PLOT 4,39: PLOT 6,39
575  PLOT 20,10: PLOT 20,11
580  COLOR= 2
585  RETURN
590  RETURN
595  PRINT  CHRS (7): VTAB 22: HTAB 13
600  PRINT "ESTA LETRA JA FOI TECLADA"
605  FOR ERT = 1 TO 800: NEXT : HTAB 13: VTAB 22
610  VTAB 22: PRINT  SPC( 25): GOTO 145
615  VTAB 22
620  HTAB I + 1: PRINT  MIDS (PS(P),I,1):C = C + 1
625  RETURN
```

IBM PC

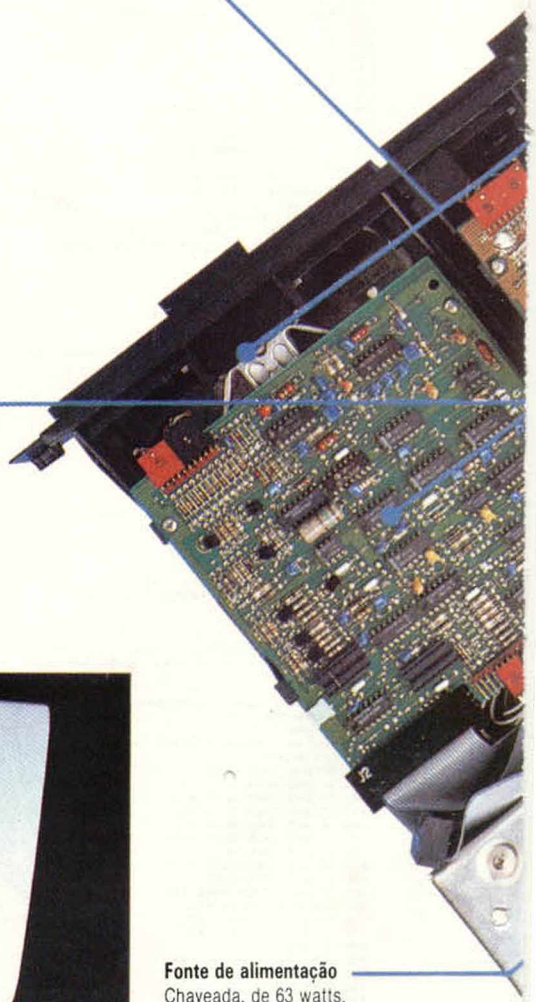
Em 1981, a International Business Machines entrou no mercado dos microcomputadores com o IBM PC (Personal Computer), que logo alcançou grande popularidade, sobretudo em aplicações comerciais.

No Brasil e em outros países, multiplicou-se a fabricação de compatíveis do IBM PC. Hoje essa máquina conta com a maior biblioteca de software disponível para qualquer micro: processadores de texto, planilhas eletrônicas, bancos de dados, geradores de gráficos e softwares integrados que combinam aplicações. Isso é possível graças a seu rápido sistema de endereçamento de 16 bits, que permite aproveitar melhor a memória e acessar mais periféricos.

O PC, máquina sólida e confiável, tem suporte técnico garantido pela IBM, a maior fabricante mundial de computadores de médio e grande porte. Seu desenho é funcional, com a CPU, o monitor e o teclado independentes; o usuário os posiciona da forma mais conveniente.

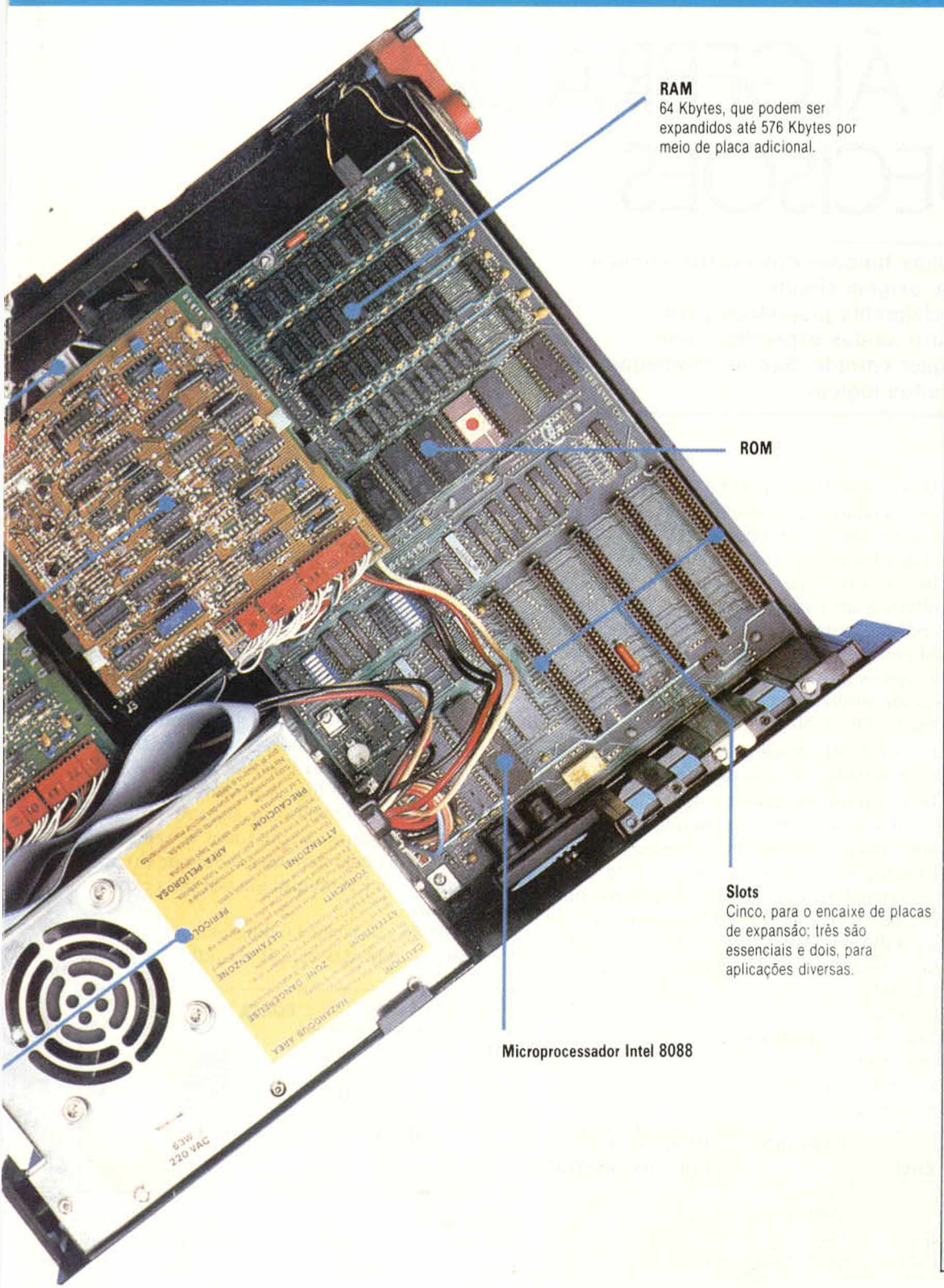
O modelo básico tem 64 Kbytes de RAM. Nele, a execução de programas mais complexos exige placas para expansão de memória, controle de vídeo e de unidades de disco e geração de gráficos.

Unidades de disco
Duas unidades de disquete flexível, de 5 1/4", com dupla face e dupla densidade e capacidade para 360 Kbytes por disco.



Placas de acionamento dos discos

Fonte de alimentação
Chaveada, de 63 watts, 110/220 VAC. Tem ventilador interno, que impede o superaquecimento.



RAM
64 Kbytes, que podem ser expandidos até 576 Kbytes por meio de placa adicional.

ROM

Slots
Cinco, para o encaixe de placas de expansão: três são essenciais e dois, para aplicações diversas.

Microprocessador Intel 8088

IBM PC

MICROPROCESSADOR

Processador central: Intel 8088
Co-processador aritmético: 8087

CLOCK

4,7 MHz

MEMÓRIA

40 K de ROM e 64 K de RAM, expansíveis até 576 Kbytes.

SISTEMA OPERACIONAL

PC-DOS, MS/DOS, CP/M-86, UNIX

VÍDEO

Monitor monocromático em fósforo verde, de 12", com capacidade multinível (várias tonalidades) e vídeo inverso. Dotado de máscara anti-reflexo. Monitor colorido opcional. Alta definição gráfica. No modo texto, a tela apresenta 25 linhas de 80 colunas.

TECLADO

Independente do gabinete da CPU, ao qual se conecta por cabo. Tem inclinação ajustável e é do tipo mecânico, com repetição automática. Possui 82 teclas, sendo dez de função, programáveis. Teclado numérico separado, tipo calculadora, e oito teclas para movimentação do cursor.

LINGUAGENS

BASIC e ASSEMBLER residentes. Diversas outras disponíveis em disco: COBOL, FORTRAN, PASCAL etc.

PERIFÉRICOS

Impressora, disco rígido, joystick, caneta óptica, tablete gráfico, sintetizador de som e de voz, mouse, recursos para teleprocessamento.

DOCUMENTAÇÃO

Completa e detalhada, incluindo manuais de operação, de linguagem BASIC e do sistema operacional DOS.

A ÁLGEBRA DAS DECISÕES

Algumas funções dos micros, como a soma, exigem circuitos especialmente projetados para produzir saídas específicas para qualquer entrada. São os chamados "circuitos lógicos".

O inglês George Boole (1815-1864) foi um dos promotores da lógica matemática. A álgebra por ele desenvolvida — ramo da matemática relacionado com a lógica do verdadeiro ou falso — forma a base teórica a partir da qual é estruturada a arquitetura do computador.

Os conceitos e as regras da lógica booleana são de fácil compreensão. Baseiam-se em três operações lógicas intimamente relacionadas com o modo como usamos, na linguagem cotidiana, as palavras E, OU e NÃO.

Observe esta afirmação: "Se o tempo estiver bom E for sábado, Júlio sairá para um passeio".

Se Júlio vai ou não passear depende de duas coisas: o dia estar bom e ser sábado. Para chegar a uma decisão a respeito, ele precisa saber se as afirmações "o dia está bom" e "é sábado" são verdadeiras ou falsas. Há quatro combinações possíveis (o dia estar bom mas não ser sábado, o dia não estar bom e não ser sábado, o dia não estar bom e ser sábado e o dia estar bom e ser sábado) e apenas uma resultará no passeio de Júlio.

O quadro que mostra todas as combinações possíveis numa série de afirmações é chamado de "tabela de validação". No caso do E lógico, a tabela de validação seria esta:

O DIA ESTÁ BOM	É SÁBADO	JÚLIO SAIRÁ PARA UM PASSEIO
FALSO	FALSO	FALSO
FALSO	VERDADEIRO	FALSO
VERDADEIRO	FALSO	FALSO
VERDADEIRO	VERDADEIRO	VERDADEIRO

Pode-se realizar um processo semelhante para ilustrar a função da operação lógica OU.

Considere esta afirmação: "Se Pedro OU Paulo puderem ir ao jogo, João também irá".

Uma de duas condições determina se João irá ou não ao jogo: Pedro poder ir ou Paulo poder ir. Como há duas condições — cada uma das quais podendo ser verdadeira ou falsa —, a tabela de validação comporta quatro combinações e somente num dos casos o sujeito não concretizará a ação.

PEDRO PODE IR	PAULO PODE IR	JOÃO IRÁ AO JOGO
FALSO	FALSO	FALSO
FALSO	VERDADEIRO	VERDADEIRO
VERDADEIRO	FALSO	VERDADEIRO
VERDADEIRO	VERDADEIRO	VERDADEIRO

A terceira operação lógica, NÃO, executa uma função muito simples.

Considere esta afirmação: "Se NÃO estiver fazendo frio, irei ao cinema".

Desta vez, a única condição a considerar é se está frio. Ela pode ser verdadeira ou falsa; assim, há apenas duas condições possíveis:

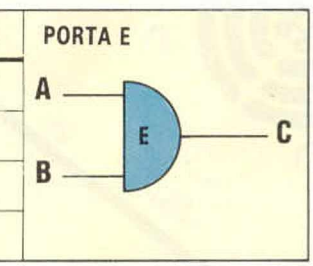
ESTÁ FRIO	IREI AO CINEMA
FALSO	VERDADEIRO
VERDADEIRO	FALSO

Portas lógicas

Os dispositivos eletrônicos simples que formam os circuitos lógicos de computador são chamados de "portas lógicas". Essas portas funcionam pela representação da condição VERDADEIRO pelo bit 1 e da condição FALSO pelo bit 0. Assim, para cada porta lógica podemos construir uma tabela de validação que mostre todas as combinações de entrada com todas as saídas resultantes. Cada porta tem um símbolo de circuito a ela associado e é representada por uma expressão booleana.

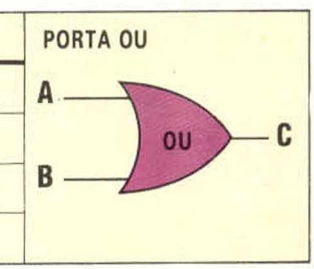
A tabela de validação e o diagrama para a porta E com as entradas A e B e saída C é:

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1



A função da porta E pode ser descrita em palavras como: "A saída será 1 se ambas as entradas forem 1, e será 0 nos demais casos". A notação booleana para essa saída de uma porta E é A·B (A vezes B).

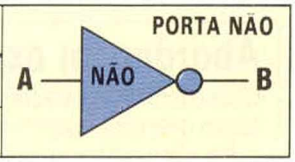
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1



A porta OU descreve-se por esta afirmação: "A saída será 1 se qualquer uma ou ambas as entradas forem 1". A expressão booleana para a saída de uma porta OU é A + B (A mais B).

Ao contrário de E e OU, a porta NÃO tem apenas uma entrada e uma saída. A tabela de validação é a mais simples das três:

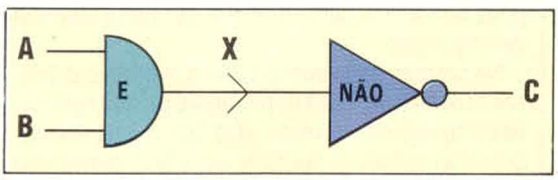
A	B
0	1
1	0



Em palavras, a porta NÃO é expressa como: "A saída será o oposto da entrada". A expressão booleana para a entrada de uma porta NÃO é A.

Combinando as portas lógicas

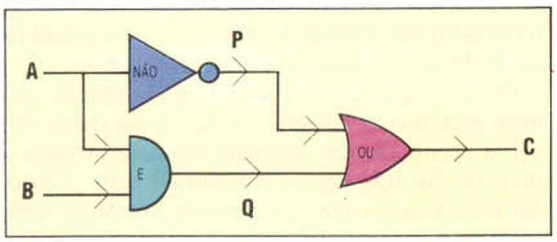
É possível unir portas lógicas para formar circuitos lógicos combinados e seqüenciais. Estes, por sua vez, podem ser arranjados para produzir a arquitetura do computador. Qualquer circuito lógico é representado por uma tabela de validação que descreve qual saída esperar para qualquer combinação possível de entradas. Observe este circuito lógico:



Nele, há duas entradas (A e B) e uma saída (C). Para auxiliar o estudante de lógica booleana a construir a tabela de validação para o circuito, a saída da primeira porta foi denominada X. Como existem duas entradas, há quatro combinações de entradas possíveis.

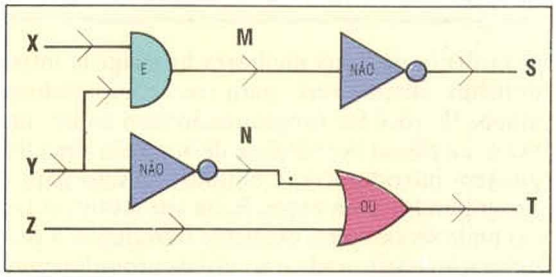
A	B	X	C
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

A saída da porta E (ou seja, X) é realizada pela porta NÃO para produzir a saída final (C). Observe, no circuito mais complexo apresentado abaixo, que, como há duas entradas, o número de combinações possíveis de entrada ainda é quatro. A segunda metade da tabela de validação (P, Q e C) corresponde à reordenação e parte de uma tabela de validação da porta OU.



A	B	P	Q	C
0	0	1	0	1
0	1	1	0	1
1	0	0	0	0
1	1	0	1	1

O uso das tabelas de validação não se limita a dois circuitos de entrada e um de saída; pode ser ampliado para qualquer circuito, como os de três entradas e duas saídas:



Como há três entradas neste circuito, devem-se considerar oito combinações possíveis:

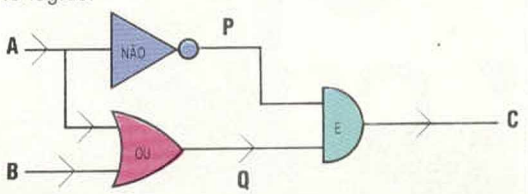
X	Y	Z	M	N	S	T
0	0	0	0	1	1	1
0	0	1	0	1	1	1
0	1	0	0	0	1	0
0	1	1	0	0	1	1
1	0	0	0	1	1	1
1	0	1	0	1	1	1
1	1	0	1	0	0	0
1	1	1	1	0	0	1

EXERCÍCIO 1

1) Construa uma tabela de validação para esta situação: "Jaime pode dirigir um carro se tiver passado em seu exame OU se estiver com um motorista habilitado".

2) Construa uma tabela de validação para esta situação: "Um programa pode ser carregado num computador se houver um gravador cassete OU uma unidade de discos E se o programa NÃO for escrito para ser processado em computador diferente".

3) Construa uma tabela de validação para este circuito lógico:



Respostas na próxima matéria desta seção.

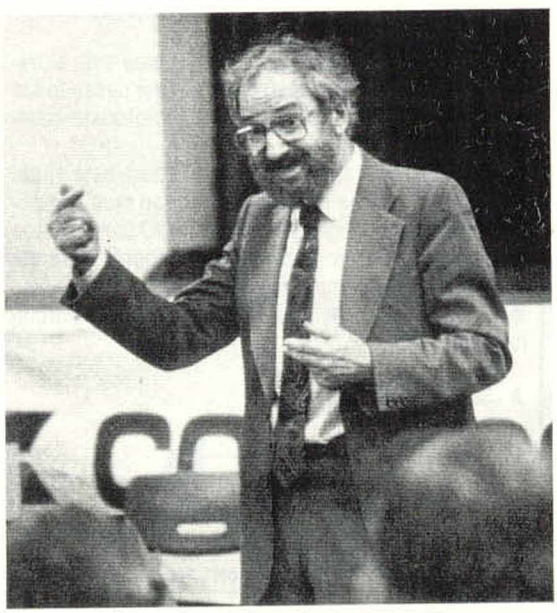
OS PASSOS DA TARTARUGA

O LOGO é considerado a linguagem ideal para desenvolver o raciocínio lógico. Descrevemos aqui a utilização de seus recursos gráficos para obter formas variadas com um mínimo de comandos.

O LOGO é uma das melhores linguagens introdutórias disponíveis para os computadores atuais. Se você faz programação com BASIC, talvez sinta pouca necessidade de aprender uma linguagem introdutória. Contudo, mesmo para o programador com experiência em BASIC, o LOGO pode servir como excelente introdução à programação estruturada e ao uso de procedimentos modulares, em vez de instruções isoladas.

Além disso, ele é encontrado em cartucho ou cassete para a maior parte dos microcomputadores. No Brasil há uma versão LOGO em português (MLOGO), cujos comandos são, por exemplo, DIREITA, FRENTE, REPITA etc. Ela pode ser executada em qualquer micro compatível com o Apple que disponha de um mínimo de 64 Kbytes de memória. Por isso, será usada em todos os nossos exemplos de programas.

Mesmo crianças pequenas (de quatro ou cinco anos) podem ser ajudadas pelo LOGO em seu aprendizado, por meio da "tartaruga", um robô mecânico ligado ao microcomputador. As tartarugas foram projetadas sobretudo para uso escolar e têm princípio de funcionamento simples. A tartaruga é equipada com duas rodas e uma caneta na parte inferior ("barriga"). A criança a instrui para que se mova sobre uma cartolina e desenhe ou não uma linha (o "rastro da tartaruga") conforme se desloca.



O criador do LOGO, Seymour Papert, numa conferência, em 1983. Papert está associado com o LOGO Computer Systems Inc. (LCSI), que fornece programas para o Sinclair Spectrum, os computadores Atari e outros.

Utilizando os comandos LOGO, a criança executa desenhos, instruindo o robô sobre o modo de formar ângulos e unir linhas. Incentivada a planejar os movimentos da tartaruga, descobre os elementos da geometria básica. Essa iniciativa individual constitui o cerne do método LOGO, fundado na idéia de que as lições aprendidas heuristicamente (por tentativa e erro, na prática) são mais bem assimiladas do que as mostradas em teoria e exemplos.

As crianças maiores (a partir de nove ou dez anos) usam a tartaruga na tela do computador, desenhando figuras complexas por meio de procedimentos modulares. Ao ensinar a tartaruga a executar movimentos na tela, a criança está na verdade programando o computador.

O LOGO, portanto, é uma linguagem que permite elaborar programas antes de se desenvolver a compreensão abstrata necessária para a maioria das linguagens computacionais. "Brincar com a tartaruga" possibilita à criança acostumar-se a controlar o computador.

Abordagem exploradora

O LOGO tem duas características fundamentais que fazem dele uma poderosa linguagem educacional.

Ele é interativo: quando você digita um comando, os resultados surgem de imediato na tela. Isso significa que é fácil fazer progresso (em particular, para crianças e principiantes), pois você confere seus resultados a cada passo.

A segunda característica vital é que o LOGO tem recursos para expansão: operações completas são manipuladas por procedimentos, ou seja, listas de instruções elementares em LOGO. Assim que se define um procedimento como um conjunto de instruções, seu nome assume a condição de um novo comando LOGO. Daí em diante, o procedimento inteiro pode ser repetido, bastando para isso digitar seu nome. Dessa forma, você pode criar seus próprios comandos, além dos que são parte inerente da linguagem.

Na programação com o LOGO, a maioria das pessoas tende a ser mais exploradora do que com outras linguagens. Às vezes, elas fazem uma abordagem meticulosa e definem um plano específico desde o começo. Em outras ocasiões, começam com um problema central e escrevem o procedimento para solucioná-lo, montando depois um programa em torno desse procedimento. É possível abordar a programação LOGO de uma maneira mais flexível, porque, em geral, há diversos meios de se chegar a um resultado.

Piaget, Minsky, Papert

O LOGO resultou do LISP, linguagem de inteligência artificial inventada no começo da década de 1960 para ajudar os computadores a trabalhar com estruturas de dados complexas. O nome LISP deriva de ser essa linguagem dedicada ao processamento de listas (LISt Processing). Sua estrutura básica de dados é uma lista, e não um string de caracteres (uma seqüência deles tratados como uma unidade) ou uma matriz numérica, como no BASIC. As funções básicas do LISP manipulam os elementos da lista, que podem ser simples símbolos ou até mesmo outras listas. A vantagem dessa abordagem é que ela processa mais facilmente os dados não numéricos

O LISP baseia-se no princípio da recorrência (recursion), que implica uma série de rotinas repetidas dentro de um programa. O resultado de cada repetição depende do resultado da repetição anterior. No caso do LISP, o item a definir é sempre uma lista.

Essas não são características acidentais do LISP: têm suas origens em pesquisas da linguagem natural e da inteligência humana.

Contudo, não se aprende facilmente o LISP, e em 1968 um grupo de pessoas associadas ao Instituto de Tecnologia de Massachusetts (MIT, Massachusetts Institute of Technology) propôs-se criar uma linguagem para crianças nele baseada.

O líder do grupo do MIT era Seymour Papert, que já havia dedicado vários anos ao estudo do desenvolvimento cognitivo infantil, com Jean Piaget (1896-1980), o mais importante psicólogo educacional de sua geração. Passando para o MIT, começou a trabalhar ao lado de Marvin Minsky, especialista em inteligência artificial. Em seu trabalho com o LOGO, Papert tentou juntar as idéias dos dois cientistas, unindo teorias do aprendizado e da inteligência artificial.

As pesquisas com o LOGO continuaram durante os anos 70, e formaram-se outros grupos dispostos a fazer experiências usando a nova linguagem. Todo esse trabalho desenvolveu-se nos departamentos de pesquisa de várias universidades, com computadores de médio e grande porte. Foi só com a chegada dos microcomputadores que o LOGO se tornou disponível para o público em geral. Essa sofisticada linguagem necessita de muita memória, tanto para o programa quanto para o espaço de trabalho. Os interpretadores LOGO encontrados nos micros requerem uma memória em torno de 30 Kbytes, e outros 8 Kbytes ou mais para a exibição dos gráficos. Tudo isso, antes de se começar a programação.

Assim, embora fosse possível implementar interpretadores BASIC simples em micros de uso doméstico a partir do momento em que eles foram comercializados, só depois que os micros com mais de 48 Kbytes de RAM passaram a ser largamente usados é que o LOGO em micros se tornou viável. Em 1980, a publicação de *Minds-*

torms (LOGO — Computadores e educação), de Seymour Papert, tirou o LOGO dos departamentos de pesquisa, atraindo a atenção de um grupo muito maior de pessoas. Nesse livro desenvolve-se uma visão de como os computadores podem ser usados na educação. A obra re-

Geometria e lógica

Quem pode se beneficiar com o aprendizado da programação em LOGO? Muitas pessoas, e até mesmo programadores experientes:

- Principiantes em computação ou programação;
- Pessoas que gostam de entreter-se com computadores e acham que usá-los deve ser uma atividade divertida;
- Aqueles que estão decepcionados por não terem adquirido um domínio suficiente em outra linguagem de programação;
- Quem se mostra interessado em aprender, ensinar e desenvolver o raciocínio lógico e matemático; e
- Usuários que desejam penetrar em áreas mais avançadas da computação, especialmente aquelas que estão relacionadas com o estudo da inteligência artificial.

Assim como o BASIC e a programação em código de máquina, o LOGO não é para ser usado por qualquer um. Especificamente, pode não ser a melhor linguagem para:

- Quem pensa que usar o computador é um "trabalho". Algumas linguagens são planejadas para trabalhar, como cavalos que puxam carroças. Mas você dificilmente escolheria um cavalo de carroça para montar e passear no campo; e
- O usuário que precisa ou espera grande velocidade no processamento das instruções pelo computador. O LOGO usa muita memória e funciona devagar na atual geração de microcomputadores. (Em programas comparáveis, o LOGO funciona com metade da velocidade do BASIC.)

No entanto, mesmo para esses grupos, o conhecimento do LOGO pode ser de grande valia para indicar a solução lógica de um problema e prepará-lo para ser programado em outra linguagem.

Versões autorizadas
Versões completas e bem documentadas do LOGO para o Commodore 64 (Terrapin-MIT), Sinclair Spectrum e os computadores Atari (LCSI). No Brasil existe a versão MLOGO, em português.

Vem aí...

... a tartaruga!

sultou da síntese de três conjuntos de idéias: teorias do desenvolvimento cognitivo, inteligência artificial e o movimento educacional que visa ao ensino centrado no aluno. Papert quer ver crianças programando computadores, e não computadores programando crianças — o que, segundo ele, acontece na maioria dos casos de "ensino com auxílio de computador".

O livro prevê o surgimento de uma nova "cultura do computador", na qual muitas idéias formais, anteriormente consideradas acima da capacidade das crianças, serão por elas dominadas com facilidade, devido à maneira pela qual elas usam o computador. Essa exploração de idéias — ativa e cooperativa (numa relação aluno-aluno e aluno-professor) — constitui a filosofia LOGO, na qual se fundamenta o uso dessa linguagem na educação.

A tartaruga

Depois de carregado, o LOGO fica em modo "imediato" e está pronto para receber e executar comandos. Na maioria das versões, esses comandos devem ser digitados com letras maiúsculas. Emita DESENHE e você verá que a tela se divide em duas partes (modo de tela dividida). A ala superior fica reservada aos gráficos e ocupa a maior parte da tela. No centro aparece a tartaruga, representada por um pequeno triângulo. A parte inferior cabe ao texto, e no momento apresentará um "?".

A tartaruga é um objeto com o qual podemos nos comunicar, transmitindo-lhe comandos. Quando se pensa na tartaruga como um objeto, fica mais fácil entender como se faz programação com ela. As coisas mais importantes a considerar são a posição da tartaruga, sua direção, e se a caneta que ela leva está virada para baixo (caso em que traça uma linha ao mover-se) ou para cima (move-se sem deixar traço). Emitindo-se o comando DESENHE, a tartaruga fica posicionada no centro da tela, virada para o alto e com a caneta abaixada. Podemos então lhe dar um comando como este:

```
FRENTE 40
```

No caso, a tartaruga subirá quarenta unidades ("passos de tartaruga") na tela, traçando uma linha. FRENTE é um comando para a tartaruga, e o número 40 é seu dado de entrada. Alguns comandos necessitam de dados, ao passo que outros não. DESENHE, por exemplo, não precisa de dados.

Um segundo comando para a tartaruga é VOLTE. VOLTE 10 instrui a tartaruga para que volte dez unidades. Portanto, FRENTE e VOLTE mudam a posição da tartaruga na tela. DIREITA e ESQUERDA, por outro lado, não mudam a posição da tartaruga, mas fazem-na girar — isto é, mudam sua direção. Estes dois últimos requerem um ângulo entre 0 e 360 graus como dado.

Experimente usar esses comandos desenhando algumas formas simples. Veja o que acontece se você instrui a tartaruga para mover-se a uma distância maior do que permite o tamanho da tela ou se usa números negativos como dados. Para começar outra vez com a tela limpa, digite DESENHE.

Quando testar os comandos, verá que a tartaruga pode entrar na outra seção da tela, e assim ficar "atrás" do texto, de modo a não ser vista. Estes comandos podem ajudá-lo:

TODATELA — permite que a área total da tela seja usada para os gráficos;
ESCREVA — remove todos os gráficos, deixando apenas o texto dos procedimentos;
MEIATELA — devolve o modo de tela dividida para gráficos e textos;
SEMCOR — faz com que a tartaruga se mova sem traçar uma linha;

Simplificação

Abreviações dos comandos apresentados nesta matéria:

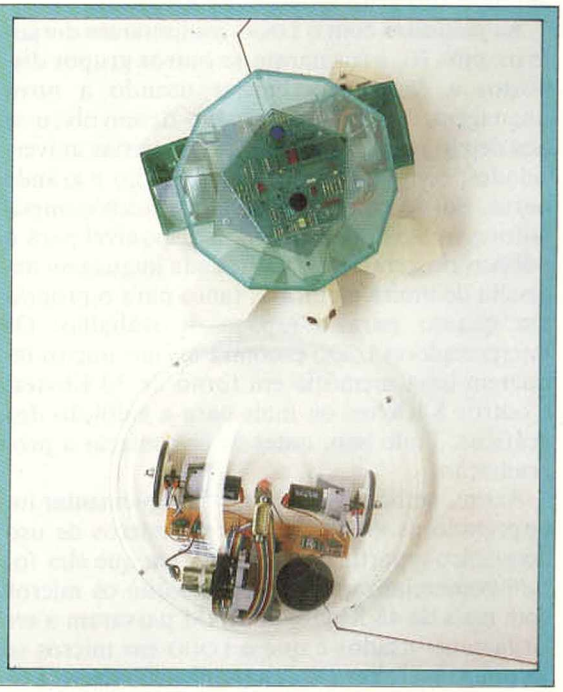
FRENTE	FR
VOLTE	VO
DIREITA	DI
ESQUERDA	ES
SEMCOR	SC
COLORIDO	CL
MOSTRE	MO

Conheça a tartaruga

Em LOGO, uma tartaruga é um robô que desenha. Ela possui rodas, controladas por motores graduais, e uma caneta retrátil. Pode ser instruída a fim de andar para a frente, para trás, para a direita e para a esquerda, e pode levantar ou abaixar sua caneta. Abaixada, a caneta desenha sobre a superfície em que a tartaruga está colocada.

Quando começaram a ser produzidas, as tartarugas tinham a forma de cúpula, como a Edinburgh *(foto de cima)*, e eram controladas pelo teclado do computador, ao qual se ligavam por um cabo paralelo. As tartarugas mais modernas funcionam por controle remoto. Existe hoje uma versão da tartaruga Edinburgh controlada por rádio. A Valiant Turtle, um robô em forma de tartaruga *(foto de baixo)*, tem uma conexão infravermelha com o computador.

Por extensão, o nome "tartaruga" também é usado para referir-se ao cursor que desenha na tela do computador, em LOGO.



COLORIDO — determina que a tartaruga deixe uma trilha enquanto se move.

Os comandos que descrevemos até agora correspondem a instruções para a tartaruga fazer o que se deseja. Mas você também pode usar o LOGO para obter informações da tartaruga. Sua direção é medida em graus, no sentido horário. A direção 0 indica que a tartaruga está virada diretamente para cima. Para descobrir a direção da tartaruga, digite:

```
MOSTRE DIRECAO
```

A posição da tartaruga na tela é definida em termos de um sistema de coordenadas, com origem no centro da tela — isto é, a tartaruga parte de um ponto onde as coordenadas *x* e *y* são iguais a 0. Você pode determinar a posição da tartaruga a qualquer momento, digitando:

```
MOSTRE CORX MOSTRE CORY
```

Teste os comandos desenhando uma figura, e então verifique a posição da tartaruga e a direção para a qual está voltada. Use esses dados para fazer a tartaruga voltar ao ponto de partida.

Mensagens de erro podem aparecer na área do texto durante suas experiências com os comandos LOGO. Se isso não acontecer, erre deliberadamente para ver o que ocorre. Digite, por exemplo:

```
FRENTE50
```

Você verá a mensagem:

```
NAO APRENDI O QUE SIGNIFICA FRENTE50
```

A razão disso é que o LOGO exige um espaço entre o comando FRENTE e o dado 50, para evitar confusão com um hipotético comando chamado FRENTE50. Você também pode obter uma mensagem de erro se digitar um comando em letras minúsculas.

O LOGO é equipado com um editor de linhas que lhe permite corrigir as instruções, se você notar um erro antes de pressionar [RETURN]. Use as teclas de movimentação do cursor para percorrer a linha até o texto incorreto. Para inserir os caracteres, basta digitá-los — o texto à direita do erro será automaticamente movimentado para acomodar os caracteres extras desejados. A tecla [ESCAPE] apaga o caractere à esquerda do cursor.

Quando a linha está correta, tecla-se [RETURN], o que faz o LOGO aceitar a nova instrução. Se você já pressionou [RETURN] antes de perceber um erro, digite [CONTROL]-P, a fim de conservar a última linha para edição. Essa função é igualmente útil quando você deseja repetir comandos.

Agora podemos tentar algo um pouco mais matemático, como um quadrado. Lembre-se de que um quadrado tem quatro lados iguais e que seus cantos formam ângulos retos (90 graus). Portanto, uma seqüência de comandos como esta produzirá o resultado desejado:

```
FR 50 DI 90
FR 50 DI 90
FR 50 DI 90
FR 50 DI 90
```

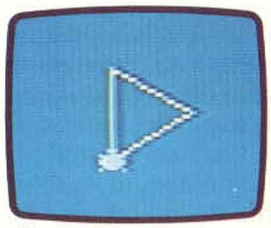
Observe que podemos abreviar os comandos e colocar mais de um comando na mesma linha. Por outro lado, você vai encontrar um problema técnico nesse ponto: seu quadrado pode ficar parecido com um retângulo. Isso se deve à desproporção entre as medidas verticais e horizontais do passo da tartaruga. Há um comando LOGO que cuida disso: use .VERTICAL seguido de um número (a definição pré-assumida é de 0.8), para modificar a proporção até que se configure um quadrado perfeito.

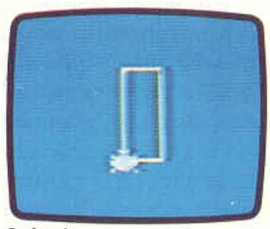
Agora, tente fazer os seguintes exercícios: desenhe um triângulo eqüilátero, um pentágono, um hexágono, diversos retângulos, um losango e um paralelogramo.

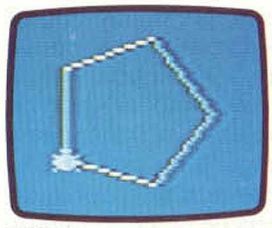
Você pode simplificar alguns dos comandos e reduzir o volume de digitação usando REPITA. Para desenhar de novo o quadrado, digite apenas:

```
REPITA 4 [FR 50 DI 90]
```

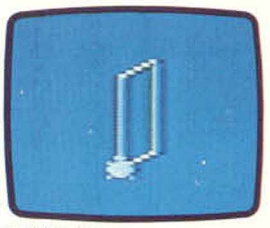
REPITA é um comando que exige dois parâmetros. O primeiro é um número que indica quantas vezes o LOGO deve fazer alguma coisa, e o segundo é uma "lista" dos comandos que devem ser obedecidos. Essa lista deve vir sempre entre colchetes. Assim, o exemplo do quadrado diz ao LOGO que ele deve repetir a seqüência FRENTE 50 DIREITA 90 quatro vezes. Agora experimente usar REPITA para simplificar a construção das figuras que você já criou, e veja se pode produzir as estrelas que aparecem nas ilustrações desta matéria.

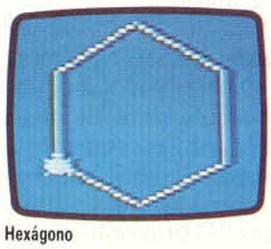

Triângulo eqüilátero


Retângulo


Pentágono

Estrela de cinco pontas


Paralelogramo


Hexágono

Estrela de dez pontas

Exercício 1
Você consegue escrever os procedimentos para criar estas formas? **Respostas na próxima matéria da série.**

PLANO DE AÇÃO

A boa programação considera aspectos como a finalidade do programa, sua utilização, os formatos de entrada e saída, os formatos dos arquivos, as rotinas e os cálculos especiais, além da estruturação lógica e modular.

Os projetos de programa são vistos pelos que estão envolvidos — analista, programador e usuário — como exercícios formais de solução de problemas. E, em geral, consideram-se apenas os aspectos técnicos — como formatar a tela, tornar os loops mais rápidos, fazer com que o programa caiba na memória, e assim por diante. No entanto, as questões já surgem no primeiro encontro do usuário com o especialista. Com raras exceções, o usuário espera do especialista não apenas a solução mas também a indicação da natureza do problema.

Junta-se a isso a freqüência com que os especialistas acreditam conhecer o problema e a solução antes mesmo de o usuário começar a esboçá-lo. Dessa má comunicação inicial resulta a descrição incompleta das necessidades do usuário. Trabalhar desse modo só pode levar a um sistema insatisfatório, que o usuário apenas aceitaria sob pressão.

Em geral, nos projetos de microcomputação, o programador, o usuário e o analista do sistema são uma só pessoa. Nesse caso, reduzem-se consideravelmente as falhas de comunicação. Apesar disso, quem acumula as funções de usuário e analista deve sempre fazer um esforço para explicar a si mesmo os problemas, as soluções e as necessidades com tanta clareza como se falasse com outra pessoa.

Um exímio aeromodelista, por exemplo, quer armazenar em seu microcomputador, acoplado a gravador cassete, descrições bem precisas dos materiais utilizados em cada modelo por ele construído. Isso lhe permitirá, no futuro, consultar seus arquivos a respeito do tipo de cola ou junta utilizado em modelos anteriores. Assim, o que o projetista teria de obter do usuário seria uma definição clara sobre:

■ **A função do programa.** Pode-se começar com uma definição vaga do objetivo, como, por exemplo: "Deve armazenar os dados de meus modelos". É necessário, porém, melhorar essa definição mediante um insistente questionamento pelo projetista. Assim, torna-se possível especificar exigências do tipo "O programa precisa armazenar minhas descrições do modelo, de sua construção e de seus materiais, à medida que forem digitadas, e exibi-las na tela quando eu teclar o nome do modelo ou alguma característica de sua construção". Isso estabelece com mais clareza as exigências do usuário e indica algumas das tarefas de programação específicas envolvidas (armazenamento de dados, pesquisa, indexação, recuperação, correção).

■ **Como o programa vai ser utilizado.** Alguns detalhes da configuração disponível ficam claros pela descrição da função, mas podem não ser suficientes. Por exemplo, se o usuário trabalhar sem monitor, não haverá interesse por exibição na tela; nesse caso, seria imprescindível uma cópia impressa dos detalhes selecionados.

■ **Qual o aspecto do programa — formatos de entrada e saída.** O programador profissional costuma trabalhar com diagramas pré-impressos que representam a tela em cada fase de entrada e de saída. Para uso pessoal, essa preocupação nem sempre é necessária, embora os gráficos de alta resolução sejam exceção. Os formatos de tela constituem um aspecto muito importante na "interface com o usuário", ou seja, o contato entre usuário e máquina. Merecem, por isso, atenção e discussão especiais, semelhantes às dedicadas aos aspectos ergonômicos mais óbvios da computação, tais como a posição dos teclados e monitores, a altura da mesa etc.

■ **Como o programa deve ser organizado — os formatos do arquivo e do programa.** O usuário pode achar que necessita estocar no mínimo cem descrições de modelo, considerando inútil menos que isso. Por outro lado, pode ser que ele jamais chegue a criar senão uma meia dúzia de modelos padronizados. O tamanho do arquivo de dados tem sérias implicações sobre seu formato e métodos de acesso. Uma pesquisa serial das seis descrições de modelo guardadas em fitas cassete poderia levar uns cinco minutos, e seria aceitável para o usuário. Mas esperar que cem modelos sejam pesquisados estaria totalmente fora de questão. Uma solução seria colocar o programa e o índice das descrições numa fita, e as descrições propriamente ditas em outras vinte fitas, classificadas por tipo de modelo.

O tamanho do programa pode também vir a ser um problema. Em alguns casos, é preciso dividi-lo em vários programas distintos, a fim de que caiba na memória disponível. Por exemplo, se a entrada de textos requer recursos comple-

xos de edição; se o programa é cheio de menus e mensagens longas; se as seções de manipulação de arquivo utilizam rotinas complicadas de pesquisa e indexação.

■ **O que o programa deve fazer — rotinas e cálculos especiais.** No exemplo dos aeromodelos não ocorriam problemas desse tipo; em outros casos, porém, são freqüentes. É possível que haja vinte maneiras boas e equivalentes de se realizar uma tarefa específica, mas o usuário poderá, ainda assim, insistir numa única solução. E mesmo que outros métodos sejam melhores, de nada adiantarão se o usuário não quiser utilizá-los.

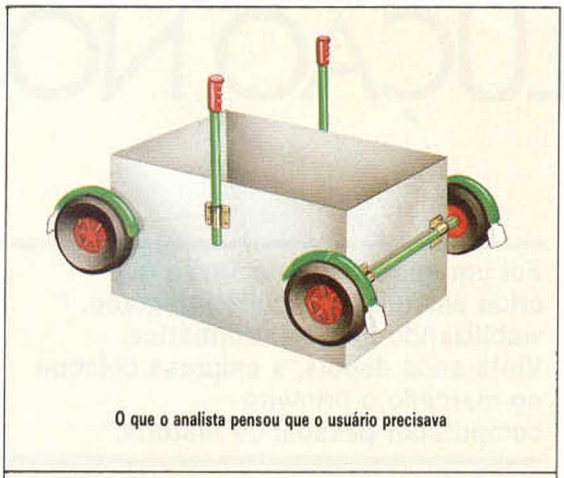
A lógica necessária

O passo seguinte é a tradução das especificações em linguagem de programação. Uma maneira prática de começar é primeiramente projetar o "diálogo" entre o usuário e o programa; em seguida, os arquivos de dados; e por último as rotinas que controlam todo o processo. Diálogo, no caso, significa troca de informações, e não, simplesmente, a entrada dos detalhes do aeromodelo e a correspondente exibição em tela. Inclui também as mensagens e os menus mostrados pelos programas, e as entradas, os comandos e as escolhas que o usuário deve digitar. Para o programa dos aeromodelos escolhe-se entre o sistema de menus e os comandos interativos. As decisões tomadas terão efeito considerável sobre a estrutura de todo o programa.

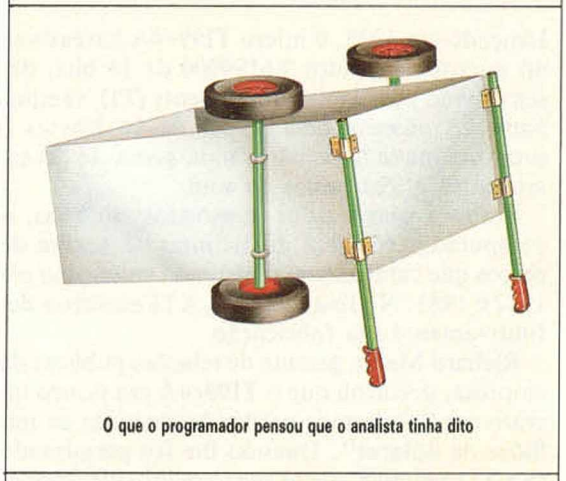
O conteúdo e a forma de comunicação devem ser considerados em detalhe, mas há uma recompensa para esse esforço: todos os dados manipulados pelo programa estarão especificados nesse momento. Isso significa que o espaço requerido para armazenamento das mensagens e advertências pode ser calculado; e torna-se possível, assim, projetar os arquivos.

No caso de arquivos longos com grandes blocos de texto, como os do programa do aeromodelo, convém dividi-los em várias fitas. Dessa forma, cada fita pode ser pesquisada com rapidez. Outro esforço útil é comprimir os dados numa codificação algorítmica, antes que sejam transcritos em fita, e, depois, decodificá-los na leitura. (Algoritmo é um conjunto de instruções que descreve um processo em termos de outros processos, básicos ou anteriormente definidos. São comuns tanto em matemática e estatística quanto em receitas de cozinha ou de tricô.)

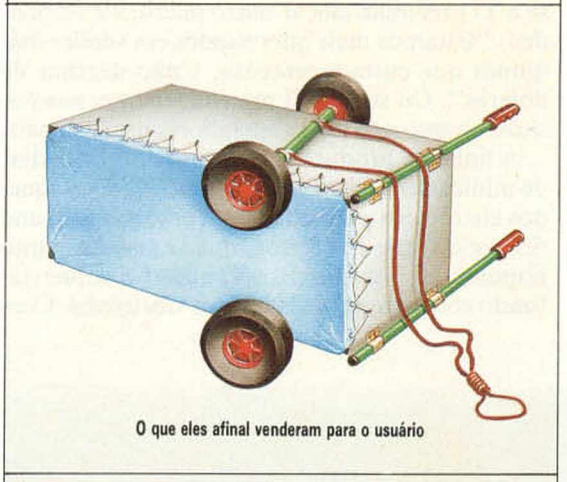
A essa altura, as funções necessárias estarão definidas. Haverá rotinas para alterar os dados já existentes e também para arquivar e classificar as novas entradas, aceitar os nomes de componentes, bem como pesquisar e exibir na tela as descrições. A essas rotinas cabe ainda a atualização de todos os índices utilizados pelo sistema. O programa deve incluir uma rotina de verificação da consistência dos dados introduzidos, a fim de rejeitar os que não sejam válidos. Todas essas opções serão apresentadas ao usuário sob a forma de menus.



O que o analista pensou que o usuário precisava



O que o programador pensou que o analista tinha dito



O que eles afinal venderam para o usuário

O que o usuário de fato queria

Descrever, definir, projetar
Se uma pouco atenta equipe de desenvolvimento de software — usuários, analistas e programadores — se propusesse a resolver o problema do transporte de cargas pesadas em jardins, é bem possível que o resultado se assemelhasse ao da ilustração. Má comunicação — entre especialistas — ainda é o maior problema a desafiar as equipes de projeto.

REVOLUÇÃO NOS CIRCUITOS

O presidente
J. Fred Bucy passou a desempenhar a função de presidente da Texas Instruments em abril de 1976.

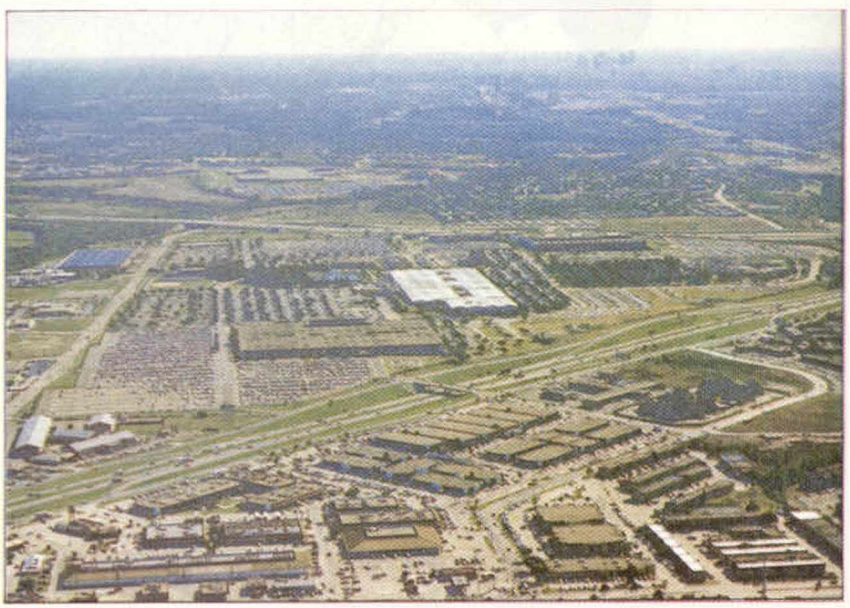
Foi um engenheiro da Texas que criou em 1958 o circuito integrado, viabilizando a microinformática. Vinte anos depois, a empresa colocou no mercado o primeiro computador pessoal da história.

Lançado em 1978, o micro TI99/4A baseava-se no microprocessador TMS9900 de 16 bits, desenvolvido pela Texas Instruments (TI). O equipamento possuía uma RAM de 16 Kbytes e proporcionava ao usuário uma gama de dezesseis cores e três canais de som.

Embora suas vendas se mostrassem boas, o computador foi uma das vítimas da guerra de preços que caracterizou o mercado americano em 1982 e 1983. No final de 1983, a TI encerrou definitivamente sua fabricação.

Richard Mann, gerente de relações públicas da empresa, declarou que o TI99/4A era pouco lucrativo e "acabamos perdendo centenas de milhões de dólares". Quando lhe foi perguntado se a TI pretendia lançar outro micro, ele respondeu: "Estamos mais interessados em vender máquinas que custam centenas, e não dezenas de dólares". Ou seja: a TI pretende manter sua posição no mercado de computadores profissionais.

A linha de produtos da empresa abrange desde minicomputadores e sismógrafos até brinquedos eletrônicos para crianças, como o Speak and Spell e o Little Professor, que se tornou muito popular, inclusive no Brasil, onde foi comercializado com o nome de Professor Corujinha. Com quinze fábricas espalhadas pelo mundo, em meados da década de 80 faturava mais de 4 bilhões de dólares por ano.



A fábrica da TI
Grande parte dos equipamentos da Texas Instruments são produzidos neste moderno complexo industrial em Dallas.

Quando foi fundada, em 1930, pelos cientistas John Karchner e Eugen McDermott, seu nome era Geophysical Service Incorporated (GSI). Karchner realizara pesquisas sobre a utilização de ondas sonoras para o cálculo da profundidade das camadas geológicas. Como pretendia vender essa sua idéia às companhias petrolíferas, resolveu abrir sua própria empresa em Dallas, no Texas.

Os anos 30 foram de contínua prosperidade para a GSI, graças ao desenvolvimento de técnicas e equipamentos de prospecção geológica. Durante a Segunda Guerra Mundial (1939-1945), esses equipamentos revelaram-se muito úteis na detecção de submarinos inimigos. Implantaram-se então um laboratório e uma linha de produção especiais na GSI. Em 1951, o setor crescera tanto que se tornou conveniente desligá-lo da GSI. A nova companhia recebeu o nome, hoje consagrado, de Texas Instruments.

No ano seguinte, a TI obteve licença dos Laboratórios Bell para a fabricação dos transistores, que acabavam de ser inventados. O primeiro rádio transistor comercial saiu de suas linhas de produção em 1954. Quatro anos depois, Jack Kilby, um de seus engenheiros, inventou o circuito integrado. Nos anos seguintes, a TI manteve-se na vanguarda das pesquisas nessa área. Kilby participou ainda do desenvolvimento da primeira calculadora eletrônica portátil, lançada em 1967.

O computador profissional da TI foi comercializado em janeiro de 1983. Baseado no processador 8088, ele pode rodar o CPM-36 e o MS-DOS. Nesse mesmo ano, a TI apresentou seu computador profissional portátil e o micro CC40, do tamanho de uma calculadora de bolso.

Na enorme lista dos componentes eletrônicos produzidos pela TI, destacam-se os chips RAM de 64 Kbytes. Os chips da Texas Instruments são encontrados em inúmeros microcomputadores. Seguindo a tendência atual, a TI também desenvolveu sua linha de processadores de 16 bits, como o TMS9900. Ao contrário de outros processadores, o TMS9900 não possui registros internos. Estes são incluídos numa memória auxiliar externa à CPU.

Segundo Richard Mann, "o TMS9900 estava um pouco à frente de seu tempo. Ele possuía muitos recursos poderosos, mas sua arquitetura não foi bem aproveitada". Posteriormente, a Texas Instruments colocou no mercado outro processador de 16 bits, o TMS99000.

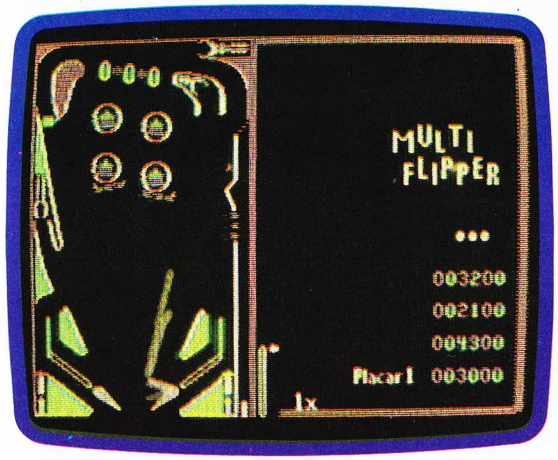
se encontra na própria raiz da cultura e é mais antigo que ela.

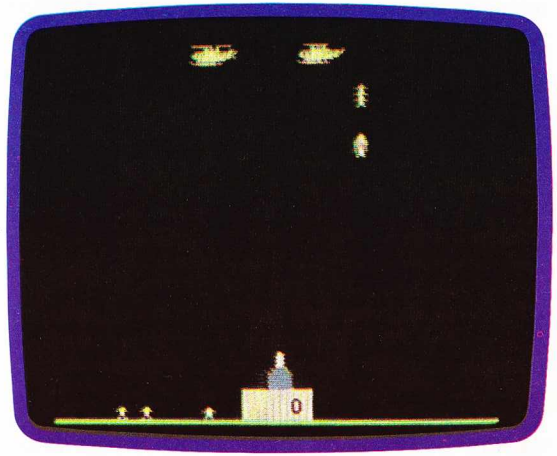
A idéia geral de jogo está presente nas fases mais primitivas da sociedade humana, e a precede: os animais convidam uns aos outros para brincar, mediante certo ritual de gestos ou atitudes. Isoladamente ou em grupos, praticam encenações, fingimentos, exibições e desafios, segundo determinadas regras. (Por exemplo: um cachorrinho se proíbe de morder, pelo menos com violência, a orelha do companheiro.) Mais importante: experimentam nisso evidente prazer e divertimento.

Assim, jogamos ou competimos por algo mais que a honra ou um prêmio material. A essência do espírito lúdico, segundo Huizinga, é ousar, correr riscos, superar a incerteza, descarregar a tensão, relaxar. Para os humanos, o jogo ultrapassa os limites de uma atividade puramente física, transcende as necessidades práticas da vida cotidiana.

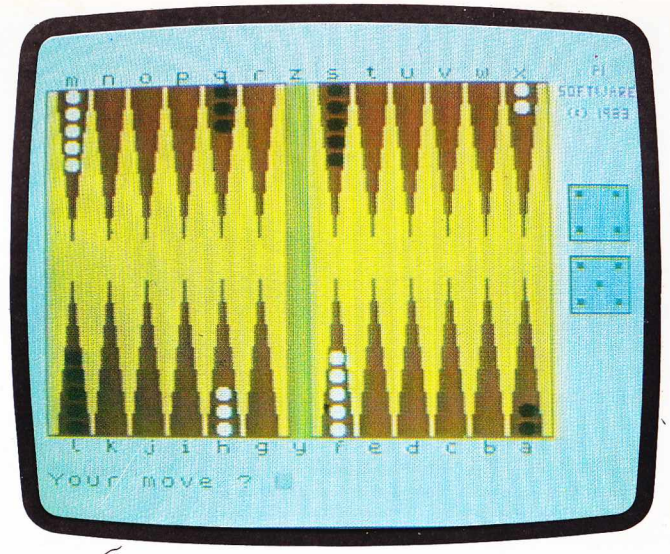
Eletrônica em cena

Nos últimos anos, os jogos eletrônicos trouxeram toda a tecnologia dos chips para a atividade lúdica. Não se pode mais negar-lhes, além da modernidade, um sentido educativo: essa modalidade de jogo prepara e adestra para a convivência — hoje absolutamente inevitável — com o admirável mundo novo dos computadores.





Simulações diversas
O Multiflipper, jogo de animação bastante sofisticado, supera em emoção seu inspirador, o fliperama. Os micros permitem maior número de opções de jogos rápidos ou de aventura, como o Sabotagem, colocando o jogador numa imensa variedade de simulações.



Pong, uma variante do pingue-pongue, foi o precursor de todos os emocionantes e originais jogos eletrônicos, que funcionam com base na microcomputação e se encontram agora em todo o mundo.

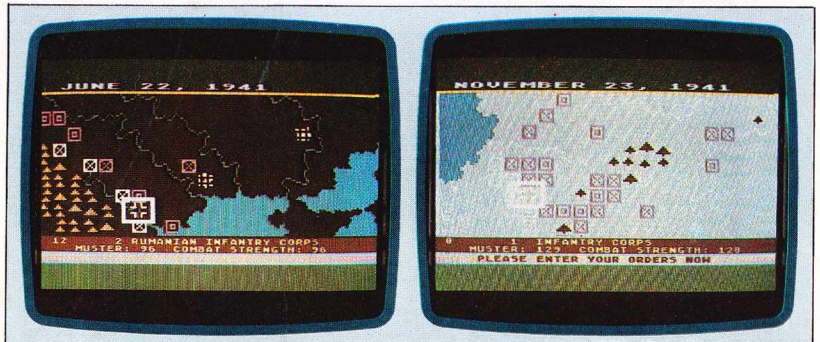
Nolan Bushnell, inventor desse primeiro fliperama, não atingiu o ponto essencial da questão: seu jogo requer duas pessoas que competem entre si; não permite que um jogador experimente sua habilidade e destreza contra a lógica da máquina.

Seis anos se passaram até o surgimento da geração seguinte de jogos eletrônicos. Só em 1977 a companhia japonesa Taito entrou em cena com Space Invaders. Neste jogo, a máquina comanda alienígenas invasores e o homem controla a arma que pode eliminá-los. O sucesso foi enorme e resultou não só em centenas de outros jogos baseados nesse princípio como em conquistas no campo da microcomputação. A necessidade de processamento rápido e grande capacidade de memória evidenciou-se primeiro entre os usuários de jogos.

O mais forte dos adversários
Desempenhando papel simultâneo de juiz imparcial e adversário implacável, os microcomputadores oferecem a possibilidade de transportar para a tela todos os jogos preferidos de mesa e de tabuleiro. Clássicos do raciocínio, como o xadrez ou o gamão, estão disponíveis no mercado para várias máquinas, em padrões visuais de alta qualidade, que produzem todos os detalhes do tabuleiro e o movimento das peças.

Na década de 1980, a coqueluche dos fliperamas foi substituída pelos potentes e baratos jogos para microcomputadores. Esses softwares proporcionam muita variedade e emoção até mesmo enquanto se está aprendendo a jogar. O aspecto lúdico da eletrônica atrai mais pessoas para a computação do que qualquer pacote de contabilidade.

A Atari, cujo êxito está fundado no jogo original de Nolan Bushnell, exemplifica o intercâmbio de jogos eletrônicos e microcomputadores. Há muito essa empresa tinha descoberto no micro um meio de lazer antes de tudo original. Além de sua série de micros, ela oferece um eficiente computador para jogos, o VCS (Vi-



Frente Oriental
Jogo histórico animado, Frente Oriental é um exemplo de fidelidade de reprodução. O jogador assume o papel do Exército alemão em 1941, com o objetivo de chegar a Moscou. Ao computador cabe a defesa do território soviético. O jogo dá idéia do decurso do tempo, associando, no mapa, a época do ano às condições climáticas da região. No outono, as florestas perdem as folhas; no inverno, os rios congelam e o chão cobre-se de neve.

deo Cartridge System), que leva para a casa do usuário, praticamente sem alterações, muitos dos jogos encontrados em equipamentos de fliperama. No Brasil, por meio da Gradiente/Polyvox, a Atari está em grande vantagem, por se tratar de importante produtora de equipamentos eletrônicos.

No entanto, outras empresas dedicadas ao lazer e ao entretenimento — como a Philips e a Sharp — estão envolvidas na mesma atividade. Adquiriram os direitos de produção de muitos jogos eletrônicos para o mercado doméstico, que podem ser utilizados em seu consoles Odissey e Intellivision, respectivamente.

Os tipos

Não se deve pensar que os jogos constituem mau emprego do micro. Eles integram o universo da computação inclusive porque divertem, instruem e desenvolvem.

Fascinantes e irresistíveis, simulam situações da vida real ou da imaginação do ser humano. Dividem-se, basicamente, em três tipos: animados, de raciocínio e de aventura.

Animados seriam os similares a Space Invaders, Jornada nas Estrelas, Ases do Volante, Fórmula 1.

Os de raciocínio são, quase sempre, adaptações para o computador de entretenimentos tradicionais, como xadrez, gamão, jogo-da-velha e dama.

Os jogos de aventura combinam elementos de animação e de raciocínio.

Essa divisão simplificadora permite a inclusão de vários outros subtipos: jogos de labirinto, de estratégia, de simulação (de vôo, das funções vitais etc.), educativos, históricos, de quebra-cabeça, de tabuleiro, de mesa (como o bridge), de investigação etc.

Em qualquer dos três tipos, a qualidade do jogo dependerá da atenção do programador em reproduzir as condições de adversidade e de probabilidade o mais fielmente possível.

Além de adquirir pacotes prontos para as diversas linhas de microcomputadores nacionais, você pode programar seus próprios jogos. Assim, ao entretenimento estimulante de disputar uma partida com a máquina somará o aprendizado de uma boa estruturação e documentação.

Nas páginas seguintes você vai encontrar programas originais especialmente desenvolvidos e selecionados para as diversas linhas de equipamento. *Jogos* apresentará um de cada vez, sempre explorando ao máximo as possibilidades de seu micro.

Bom divertimento!

HOMEM LÚDICO

A Terceira Guerra Mundial
Na linha do entretenimento, desenvolveram-se complexos sistemas de hardware e software que simulam todas as condições de um conflito, como mudanças de tática do inimigo, distribuição das forças, suprimentos, clima, e outras variáveis pertinentes às operações militares. O jogador, de repente, pode encontrar-se em meio a uma terrível batalha aérea ou num confronto entre as forças da OTAN e as do Pacto de Varsóvia.

Você liga seu microcomputador ao sistema telefônico, para acessar informações de outro computador. Na tela, aparece um jogo que você desconhece: Guerra Termonuclear. Fascinado, você entra na disputa, que envolve bases nucleares americanas contra soviéticas.

Só muito mais tarde você descobre que quebrou acidentalmente a senha do computador central da OTAN (Organização do Tratado do Atlântico Norte) e deu início ao mais terrível dos jogos: a Terceira Guerra Mundial.

Este é o tema do inquietante filme *Wargames (cena acima)*. Baseada na enorme difusão dos jogos eletrônicos de computador e na crescente sofisticação dos micros, a história mostra a correlação entre a atividade lúdica, o faz-de-conta e a vida real.

Por que o homem brinca

O fenômeno do jogo já preocupou muitos estudiosos, mas foi o historiador holandês Johan Huizinga, no livro *Homo ludens (Homem lúdico)*, quem melhor o definiu: longe de ser apenas uma entre outras atividades da vida civilizada, o jogo